

# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



### THESIS

**SIMULATION VALIDATION AND FLIGHT  
PREDICTION OF UH-60A BLACK HAWK  
HELICOPTER/SLUNG LOAD CHARACTERISTICS**

by

Peter H. Tyson

March 1999

Thesis Co-Advisors:

E. Roberts Wood  
Mark B. Tischler

19990504 107

Approved for public release; distribution is unlimited.

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

<b>1. AGENCY USE ONLY (Leave blank)</b>		<b>2. REPORT DATE</b> March 1999	<b>3. REPORT TYPE AND DATES COVERED</b> Master's Thesis
<b>4. TITLE AND SUBTITLE</b> SIMULATION VALIDATION AND FLIGHT PREDICTION OF UH-60A BLACK HAWK HELICOPTER/SLUNG LOAD CHARACTERISTICS			<b>5. FUNDING NUMBERS</b>
<b>6. AUTHOR(S)</b> Tyson, Peter H.			
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Naval Postgraduate School Monterey, CA 93943-5000			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b>			<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>
<b>11. SUPPLEMENTARY NOTES</b> The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited.			<b>12b. DISTRIBUTION CODE</b>
<b>13. ABSTRACT (Maximum 200 words)</b> Helicopter/slung load systems are two body systems in which the slung load adds its rigid body dynamics, aerodynamics, and sling stretching dynamics to the helicopter. The slung load can degrade helicopter handling qualities and reduce the flight envelope of the helicopter. Confirmation of system stability parameters and envelope is desired, but flight test evaluation is time consuming and costly. A simulation model validated for handling quality assessments would significantly reduce resources expended in flight testing while increasing efficiency, productivity, and safety by aiding researchers, designers, and pilots to understand factors affecting helicopter-slung load handling qualities. This thesis describes a comprehensive dynamics and aerodynamics model for slung load simulation, obtained by integrating the NASA Ames Gen Hel UH-60A simulation with slung load equations of motion. Frequency domain analysis is used to compare simulation to flight test frequency responses and key system stability parameters. Results are given for no load, a 4K lb Block, and a 4K lb CONEX load. Handling quality parameters, stability margins, and load pendulum motion roots for cases without load aerodynamics and with static wind tunnel data were compared. Results illustrated state-of-the-art simulation modeling of helicopter/slung load dynamics and its accuracy in predicting key dynamic parameters of interest.			
<b>14. SUBJECT TERMS</b> Air Vehicles, Helicopter/Slung Load, External Load, Handling Quality, Stability Margin, Flight Simulation			<b>15. NUMBER OF PAGES</b> 303
			<b>16. PRICE CODE</b>
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b> UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std Z39-18



Approved for public release; distribution is unlimited

**SIMULATION VALIDATION AND FLIGHT PREDICTION OF UH-60A BLACK  
HAWK HELICOPTER/SLUNG LOAD CHARACTERISTICS**

Peter H. Tyson  
Lieutenant, United States Navy  
B.S., United States Naval Academy, 1991

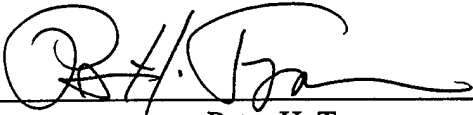
Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING**

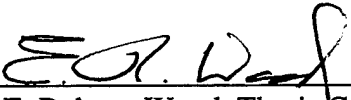
from the

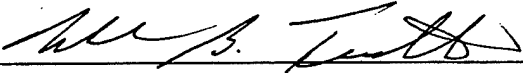
**NAVAL POSTGRADUATE SCHOOL  
March 1999**

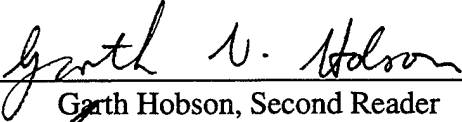
Author:

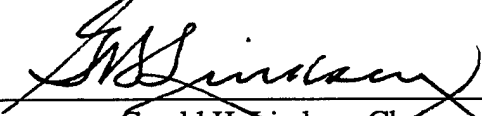
  
Peter H. Tyson

Approved by:

  
E. Roberts Wood, Thesis Co-Advisor

  
Mark B. Tischler, Thesis Co-Advisor

  
Garth Hobson, Second Reader

  
Gerald H. Lindsey, Chairman  
Department of Aeronautics and Astronautics





## ABSTRACT

Helicopter/slung load systems are two body systems in which the slung load adds its rigid body dynamics, aerodynamics, and sling stretching dynamics to the helicopter. The slung load can degrade helicopter handling qualities and reduce the flight envelope of the helicopter. Confirmation of system stability parameters and envelope is desired, but flight test evaluation is time consuming and costly. A simulation model validated for handling quality assessments would significantly reduce resources expended in flight testing while increasing efficiency, productivity, and safety by aiding researchers, designers, and pilots to understand factors affecting helicopter-slung load handling qualities.

This thesis describes a comprehensive dynamics and aerodynamics model for slung load simulation, obtained by integrating the NASA Ames Gen Hel UH-60A simulation with slung load equations of motion. Frequency domain analysis is used to compare simulation to flight test frequency responses and key system stability parameters.

Results are given for no load, a 4K lb Block, and a 4K lb CONEX load. Handling quality parameters, stability margins, and load pendulum motion roots for cases without load aerodynamics and with static wind tunnel data were compared. Results illustrated state-of-the-art simulation modeling of helicopter/slung load dynamics and its accuracy in predicting key dynamic parameters of interest.



## TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	HELICOPTER/SLUNG LOAD OPERATIONS .....	1
1.	Safety .....	2
2.	Cost .....	3
B.	UNITED STATES / ISRAELI MEMORANDUM OF AGREEMENT .....	4
C.	PURPOSE OF RESEARCH .....	4
D.	SCOPE OF WORK PERFORMED .....	5
II.	NASA FLIGHT TESTING .....	9
A.	TEST AIRCRAFT DESCRIPTION.....	9
B.	FLIGHT TEST LOADS.....	12
C.	FLIGHT TEST DATA ANALYSIS .....	14
1.	Handling Quality .....	15
2.	Stability Margin .....	17
3.	Load Pendulum Motion .....	19
III.	SLUNG LOAD SIMULATION.....	21
A.	GEN HEL ADVANCED ROTOR/HELICOPTER MODEL .....	21
B.	SLUNG LOAD DYNAMICS .....	23
C.	MAIN ROTOR WAKE MODEL .....	25
1.	Wake Geometry .....	25
2.	Downwash Velocity .....	27
D.	LOAD AERODYNAMIC FORCES AND MOMENTS .....	28
1.	Drag Force Only Estimation .....	29
2.	CONEX Static Aerodynamics.....	30
3.	Simulation Static Aerodynamic Model .....	33
E.	GEN HEL/SL SIMULATION .....	33
1.	Computer-Generated Control Sweep Input.....	35
2.	Control Feedback Loop.....	35
3.	Data Collection and Processing Automation .....	36
IV.	SIMULATION VALIDATION .....	37
A.	VALIDATION METHODOLOGY .....	37
B.	NO-LOAD SIMULATION FIDELITY .....	38
1.	Handling Quality .....	39
2.	Stability Margins.....	40
C.	4K BLOCK LOAD .....	41
1.	Handling Quality .....	41
2.	Stability Margins.....	43
3.	Load Pendulum Roots.....	44

D.	4K CONEX LOAD.....	46
1.	Handling Quality.....	46
2.	Stability Margins.....	47
3.	Load Pendulum Roots.....	47
E.	DATA TRENDS.....	48
V.	CONCLUSIONS.....	51
A.	EVALUATION OF GEN HEL/SL.....	51
1.	Simulation Discrepancies.....	51
2.	Handling Quality Prediction.....	52
3.	Stability Margin Prediction.....	52
4.	Load Pendulum Mode Prediction.....	52
B.	AREAS FOR FUTURE INVESTIGATION.....	52
1.	Model Development and Correction.....	53
2.	Envelope Expansion.....	53
3.	Load Instability and Stabilization Methods.....	54
	APPENDIX A: THEORETICAL BACKGROUND.....	55
A.1.	ONE AND TWO DEGREE OF FREEDOM APPROXIMATIONS.....	55
A.2.	SLUNG LOAD EQUATIONS OF MOTION.....	58
A.3.	COORDINATE SYSTEM TRANSFORMATIONS.....	61
1.	Helicopter Body Axes To Inertial Frame.....	62
2.	Load Body Axes To Inertial Frame.....	63
3.	Main Rotor Tip Path Plane Axes To Helicopter Body Axis.....	64
4.	Main Rotor Hub (Shaft) To Helicopter Body Axes.....	65
5.	Main Rotor Wake To Helicopter Body Axes.....	65
	APPENDIX B. GEN HEL/SL - OPERATION AND PROGRAM LISTINGS.....	68
B.1.	GEN HEL/SL PROGRAM FILES.....	69
B.2.	SUBROUTINE conexaero.....	74
B.3.	DATAFILE ghsl.dat.....	84
B.4.	PROGRAM ghsl_dat.....	86
B.5.	SUBROUTINE ghsl_init.....	96
B.6.	SUBROUTINE ghslmc.....	99
B.7.	SUBROUTINE ghslmc_ic.....	114
B.8.	SUBROUTINE nrt_unc3_in(NPTS).....	123
B.9.	SUBROUTINE nrt_unc3_out(NRESET).....	125
B.10.	SUBROUTINE pilot(NRESET).....	128
B.11.	SUBROUTINE wake.....	130
B.12.	COMMON FILE slvars.cmn.....	136

APPENDIX C. DATA COLLECTION AND PROCESSING AUTOMATION .....	143
C.1. DATA COLLECTION AUTOMATION.....	143
C.2. DATA PROCESSING AUTOMATION .....	147
1. Handling Qualities Analysis .....	147
2. Stability Margin Analysis .....	151
3. Load Characteristics Analysis.....	154
APPENDIX D. NO EXTERNAL LOAD DATA .....	157
APPENDIX E. 4K BLOCK EXTERNAL LOAD DATA .....	185
APPENDIX F. 4K CONEX LOAD DATA .....	225
REFERENCES.....	279
INITIAL DISTRIBUTION LIST.....	283



## LIST OF FIGURES

Figure 1. UH-60A Conducting Slung Load Operations.....	2
Figure 2. UH-60A General Configuration (After Ref. [7]).....	10
Figure 3. Helicopter and Load Data Acquisition System.....	11
Figure 4. Handling Quality Determination (After Ref. [1]).....	16
Figure 5. Stability Margin Determination.....	17
Figure 6. Simplified Helicopter Control Model.....	18
Figure 7. Load Pendulum Mode Determination With NAVFIT.....	19
Figure 8. NASA Ames Gen Hel UH-60A Components (After Ref. [15]).....	22
Figure 9. General Multi-Cable Sling Configuration.....	24
Figure 10. Wake Coordinate System.....	26
Figure 11. Wake Velocity Components.....	28
Figure 12. CONEX Wind Tunnel Model Internal Mechanism (From Ref. [4]).....	30
Figure 13. Technion Low-Speed Wind Tunnel.....	31
Figure 14. Wind Tunnel Banana Arm (From Ref. [4]).....	32
Figure 15. CONEX Static Aerodynamic Coefficients.....	34
Figure 16. Gen Hel/SL Simulation Architecture.....	34
Figure 17. Computer Generated Frequency Sweep.....	36
Figure 18. Validation Methodology.....	37
Figure 19. No Load Configuration Handling Quality Results.....	40
Figure 20. No Load Configuration Stability Margin Results.....	41
Figure 21. 4K Block Lateral Axis Handling Quality Results.....	43
Figure 22. 4K Block Longitudinal Axis Handling Quality Results.....	43
Figure 23. 4K Block Stability Margin Results.....	44
Figure 24. 4K Block Load Characteristics Results.....	45
Figure 25. 4K CONEX Lateral Axis Handling Qualities Results.....	46
Figure 26. 4K CONEX Stability Margin Results.....	47
Figure 27. 4K CONEX Load Characteristics Results.....	48
Figure 28. Effect of Load Weight on Handling Quality.....	49
Figure 29. Effect of Load Weight on Stability Margin.....	50

## APPENDIX A

Figure A.1. One and Two DOF Pendulum Models.....	56
Figure A.2. Slung Load Equations of Motion (After Ref. [12]).....	60
Figure A.3. Helicopter Body Axis Euler Angles.....	63
Figure A.4. Load Body Axis Euler Angles.....	64
Figure A.5. Main Rotor Wake Axis Directions.....	67

## APPENDIX D

Figure D.1. No Load Handling Quality, Lateral Axis.....	160
Figure D.2. No Load Handling Quality Determination, Lateral Axis.....	162
Figure D.3. No Load Handling Quality, Longitudinal Axis.....	166
Figure D.4. No Load Handling Quality Determination, Longitudinal Axis.....	168
Figure D.5. No Load Stability Margin, Lateral Axis.....	172
Figure D.6. No Load Stability Margin Determination, Lateral Axis.....	174



Figure D.7. No Load Stability Margin, Longitudinal Axis .....	178
Figure D.8. No Load Stability Margin Determination, Longitudinal Axis .....	180

## APPENDIX E

Figure E.1. 4K Block Handling Quality, Lateral Axis .....	190
Figure E.2. 4K Block Handling Quality Determination, Lateral Axis .....	192
Figure E.3. 4K Block Handling Quality, Longitudinal Axis .....	196
Figure E.4. 4K Block Handling Quality Determination, Longitudinal Axis .....	198
Figure E.5. 4K Block Stability Margin, Lateral Axis .....	202
Figure E.6. 4K Block Stability Margin Determination, Lateral Axis .....	204
Figure E.7. 4K Block Stability Margin, Longitudinal Axis .....	208
Figure E.8. 4K Block Stability Margin Determination, Longitudinal Axis .....	210
Figure E.9. 4K Block Load Motion, Lateral Axis .....	214
Figure E.10. 4K Block Load Motion Determination, Lateral Axis .....	216
Figure E.11. 4K Block Load Motion, Longitudinal Axis .....	219
Figure E.12. 4K Block Load Motion Determination, Longitudinal Axis .....	221

## APPENDIX F

Figure F.1. 4K CONEX Handling Quality, Lateral Axis .....	230
Figure F.2. 4K CONEX Handling Quality Determination, Lateral Axis .....	233
Figure F.3. 4K CONEX Handling Quality, Longitudinal Axis .....	238
Figure F.4. 4K CONEX Handling Quality Determination, Longitudinal Axis .....	241
Figure F.5. 4K CONEX Stability Margin, Lateral Axis .....	246
Figure F.6. 4K CONEX Stability Margin Determination, Lateral Axis .....	249
Figure F.7. 4K CONEX Stability Margin, Longitudinal Axis .....	254
Figure F.8. 4K CONEX Stability Margin Determination, Longitudinal Axis .....	257
Figure F.9. 4K CONEX Load Motion, Lateral Axis .....	262
Figure F.10. 4K CONEX Load Motion Determination, Lateral Axis .....	265
Figure F.11. 4K CONEX Load Motion, Longitudinal Axis .....	270
Figure F.12. 4K CONEX Load Motion Determination, Longitudinal Axis .....	273

## LIST OF TABLES

Table 1. ARMY 748/UH-60A General Specifications (After Refs. [7], [8], and [9]) .....	10
Table 2. Load Configuration Geometry and System Mass Parameters.....	13
Table 3. Load Pendulum Natural Frequency Estimates .....	14
Table 4. Flight Test Matrix .....	14
Table 5. No Load Simulation Correction Factors .....	39
Table D.1. Helicopter Response Summary: No Load Configuration.....	158
Table E.1. Helicopter Response Summary: 4K Block Load Configuration.....	186
Table E.2. Load Motion Parameters: 4K Block Load Configuration.....	189
Table F.1. Helicopter Response Summary: 4K CONEX Load Configuration.....	226
Table F.2. Load Motion Parameters: 4K CONEX Load Configuration.....	229



## LIST OF SYMBOLS, ACRONYMS, AND ABBREVIATIONS

<b>1</b>	Helicopter Body
<b>2</b>	Load Body
<b>a</b>	Helicopter Cargo Hook
<b>C</b>	Damping Constant, lbs/ft/sec
<b>D/q</b>	Drag Parameter, Drag Force over Dynamic Pressure
<b>DL</b>	Disk Loading, lbs/ft <sup>2</sup>
<b>e(s)</b>	Control Input plus SAS Output (LaPlace domain)
<b>FA</b>	Aerodynamic Force Vector
<b>FC</b>	Hook Force Vector
<b>f(s)</b>	SAS Output plus Linkage Dynamics (LaPlace domain)
<b>f<sub>SAS</sub>(s)</b>	SAS Output without Linkage Dynamics (LaPlace domain)
<b>g</b>	Gravitational Constant
<b>I<sub>xx</sub>, I<sub>yy</sub>, I<sub>zz</sub></b>	Moments of Inertia about the x, y, and z axes
<b>J</b>	Inertia Matrix
<b>K</b>	Spring Constant, lb/ft
<b>l</b>	Sling Length, ft
<b>l<sub>0</sub></b>	Sling Cable Initial Length, ft
<b>MA</b>	Aerodynamic Moment Vector
<b>MC</b>	Hook Moment Vector
<b>m</b>	Number of Cables in Sling
<b>m<sub>1</sub></b>	Mass of Helicopter
<b>m<sub>2</sub></b>	Mass of Load
<b>R</b>	Main Rotor Radius, ft
<b>R</b>	Position Vector
<b>r(s)</b>	Pilot Input (LaPlace domain)
<b>r/R</b>	Non-dimensional Horizontal Load Position Relative to Main Rotor Hub
<b>S</b>	Skew-Symmetric Matrix
<b>T</b>	Coordinate Transformation Matrix
<b>u, v, w</b>	Velocity Components
<b>V</b>	Velocity Vector
<b>V'</b>	Near-wake Velocity Vector
<b>V''</b>	Far-wake Velocity Vector
<b>x<sub>0</sub>, y<sub>0</sub></b>	Wake Core Offsets in the x <sub>w</sub> - and y <sub>w</sub> -axis directions
<b>z</b>	Helicopter Hook-to-c.g. Offset, ft
<b>z/R</b>	Non-dimensional Vertical Load Position Relative to Main Rotor Hub
<b>α</b>	Angle of Attack, deg
<b>β</b>	Sideslip Angle, deg
<b>δα, δb, δc, δp</b>	Control Inputs in Lateral, Longitudinal, Collective, and Yaw Axes
<b>φ, θ, ψ</b>	Euler Rotation Angles
<b>ρ</b>	Air Density, lbs/ft <sup>3</sup>
<b>τ</b>	Time for Wake to Travel from Main Rotor Hub to 1.5R, sec
<b>τ<sub>pd</sub></b>	HQ Phase Delay, sec
<b>ω<sub>n</sub></b>	Natural Frequency, rad/sec
<b>ω<sub>BW</sub></b>	HQ Bandwidth Frequency, rad/sec
<b>ω<sub>l35°</sub></b>	HQ Bandwidth Frequency from -135° Phase Crossing, rad/sec

$\omega_{dB}$	HQ Bandwidth Frequency from 6 dB Gain, rad/sec
$\zeta$	Damping Ratio
*	Center of Gravity Location
ADAS	Aircraft Data Acquisition System
ADS	Aeronautical Design Standard
AFDD	Aeroflightdynamics Directorate, NASA Ames Research Center
AOA	Angle of Attack
c.g.	Center of Gravity
CIFER®	Comprehensive Identification from Frequency Responses
COMPOSITE	Composite Frequency Analysis
CONEX	Container Express, 6 x 6 x 8 ft Cargo Container
FAA	Federal Aviation Administration
FRESPID	Frequency Response Identification
FT	Flight Test Data
Gen Hel/SL	NASA Ames (Sikorsky) General Helicopter/Slung Load Simulation
GH	Gen Hel/SL simulation data
GM	Gain Margin
HQ	Handling Quality
LC	Load Characteristics
LS	Least Squares
MILVAN	Military Van, 8 x 8 x 40 ft Cargo Container
MISOSA	Multi-Input, Single Output Spectral Analysis
MOA	Memorandum of Agreement
NASA	National Aeronautics and Space Administration
NAVFIT	Program for Transfer Function Curve-Fitting
PCM	Pulse Code Modulation
PM	Phase Margin
SAS	Stability Augmentation System
SM	Stability Margin
TPP	Tip Path Plane
VMS	NASA Vertical Motion Simulator

## ACKNOWLEDGMENTS

The author would like to acknowledge the financial support of the Army/NASA Rotorcraft Division, for allowing the purchase of equipment used in this thesis, as well as providing the necessary travel and lodging funds for a three month internship at Ames Research Center, many trips between Monterey and San Jose, and for travel to Haifa, Israel to meet with Israeli counterparts in the US/Israel MOA.

Special thanks go to my thesis advisors, Dr. E. Roberts Wood for his valuable insight on helicopter dynamics, and Dr. Mark Tischler for expert guidance on frequency domain testing and for making me a part of the research team at Ames. Also to Mr. Luigi Cicolani, the lead researcher on the task, for many long hours of instruction and discussion of slung load dynamics.

Appreciation and thanks is due to all of the Aeroflightdynamics Branch at NASA Ames, who from day one made me feel at home and continued to encourage and support me all the way through. Their contributions are too numerous to list, and there are too many individuals involved to list here. Additionally, Aviv Rosen and Danny Levine at the Technion are to be thanked for providing wind tunnel results as well as hospitality during my visit to their country. Lastly, I thank my wife and son, who gave me the time and support to do this project, and were willing to spend time apart to get the job done.



## **I. INTRODUCTION**

Helicopter/slung load systems are two body systems in which the slung load adds its rigid body dynamics and aerodynamics to that of the helicopter. The slung load can degrade the handling qualities of the helicopter and reduce the flight envelope of the combined system below that of the helicopter alone. Additionally, the effects of the load vary significantly among the load/sling combinations that a utility helicopter will encounter during its operational life. Therefore confirmation of system stability handling qualities and envelope is desired, but flight test evaluation of these parameters is time consuming and costly, as indicated by the millions of dollars spent by the helicopter industry to certify cargo hooks by proving safe operation throughout the flight envelope. A simulation model validated over the range of frequencies of interest in handling quality would significantly reduce the resources expended in flight testing while increasing efficiency, productivity, and safety by aiding researchers, designers, and pilots to understand the factors affecting helicopter-slung load handling qualities.

### **A. HELICOPTER/SLUNG LOAD OPERATIONS**

The helicopter's vertical flight capability has provided a way for aviation to become vitally involved in activities that fixed wing aircraft cannot participate in. For example, rotary wing aircraft have provided airborne rescue platforms, while hovering above a survivor in the water or on land. Vertical landing capability has been exploited to provide access to places normally closed to fixed wing aircraft which require a large flat landing surface. Among the unique capabilities of the helicopter are those exploited in slung load operations, in which an object can be picked up external to the helicopter, conveyed to a remote or inaccessible spot, and deposited with precision in the desired location.



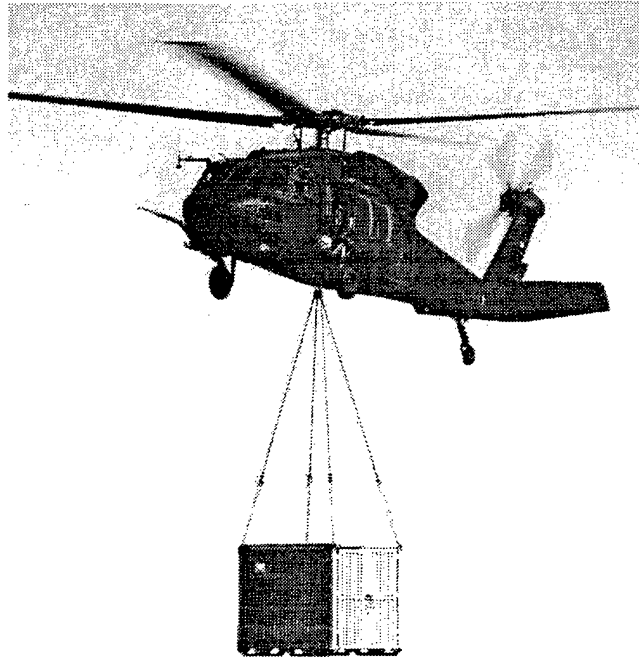


Figure 1. UH-60A Conducting Slung Load Operations

The slung load mission for the helicopter has been extended to civilian uses such as construction, logging, and fire-fighting, and to military uses such as the conveying of ordinance, supplies, or fuel bladders. These examples are just a few of the current uses and serve to give an idea of the diversity of the loads carried by helicopters in slung load operations. Figure 1 shows an Army UH-60A conducting slung load operations during testing at NASA Ames Research Center, Moffett Field, California.

## **1. Safety**

When conducting helicopter external load operations, consideration must be given to desired and actual performance characteristics of the helicopter/slung load configuration. After using analysis tools to compute the theoretical maximum weight and size of the slung load, careful flight testing is conducted to ensure that the helicopter is physically capable of safely

carrying the intended load. This extensive testing is intended to evaluate the controllability of the combination. The controllability of helicopters has been quantified by the US Army in ADS-33D (Ref. [1]) by comparing frequency responses to pilot ratings using two important parameters, the handling quality and the stability margin. For purposes of slung load system analysis a third parameter is added, the load pendulum motion. These three parameters are used to evaluate stability characteristics of a configuration, and can give a relative comparison between different helicopter/load and maneuver combinations.

Of particular interest in the study of slung load stability is the phenomenon of air resonance, a load-airframe-rotor dynamic interaction which has potentially catastrophic results. This can occur in helicopters that have fully articulated, bearingless, or hingeless main rotors. If the regressive lead-lag frequency of the main rotor blades is near the frequency of oscillation for the slung load, mutual interference between the two modes can cause the blade in-plane motion to build. The blades are then displaced out of pattern, resulting in unbalanced centrifugal forces in the rotor head. This unbalance can rapidly build and place excessive forces on the helicopter, leading to rotor head damage and possibly to loss of the helicopter. (Ref. [2])

In slung load operations, the pilot has no choice but to release the load before the destructive forces can build up. Similar mechanical instability has occurred in the case of the military CH-53 helicopter, where the fuselage bending mode for the tail section was excited by the presence of the load. The instability led to the failure of the tail section, loss of the aircraft, and loss of life.

## **2. Cost**

There are several costs associated with slung load operations that would be reduced by a complete understanding of the dynamics involved with the helicopter/load system. First, the financial impact from loss of helicopter and/or load could be averted. Secondly, the lower

throughput resulting from restrictive flight envelopes and repeated trips required by smaller size loads can be increased. Finally, the large price associated with flight evaluation of new load configurations may be reduced through the use of a computer simulation tool to explore the flight envelope and identify areas of concern prior to flight testing.

## **B. UNITED STATES / ISRAELI MEMORANDUM OF AGREEMENT**

Cooperative research into the mechanics of helicopter/slung load systems is currently performed under an agreement between the United States and Israel, under a Memorandum of Agreement (MOA) begun in October 1986. The MOA on "Rotorcraft Aeromechanics and Man-Machine Integration" covers many helicopter-related topics, including "Task VIII: Flight Mechanics of Helicopter/Sling-Load Systems." The other active programs contained in the MOA include (Refs. [3] and [4]):

- Task IV: Unsteady Flow Control
- Task VII: Human Vision Modeling
- Task IX: Human Performance Modeling in MIDAS
- Task X: High Fidelity Flight Mechanics Modeling for Simulation in a Workstation Environment

The objectives for Task VIII are:

- Develop a numerical simulation modeling technique which can accurately estimate helicopter/slung load envelopes.
- Develop and demonstrate flight test methods for rapidly verifying helicopter/slung load envelopes.
- Use the simulation to optimize slung load operations.

## **C. PURPOSE OF RESEARCH**

In support of the objectives for Task VIII, the purpose of this research effort was twofold: (1) to provide a validated helicopter/slung load simulation for existing configurations; and (2) to provide a dynamic prediction tool for new configurations.

The first goal served three additional functions. A validated simulation model could then be used to drive a motion-based flight simulator used for realistic pilot training, providing the correct “feel” of external load operations. Secondly, the model could be used for load certification and configuration experimentation. Finally, the flight envelopes for existing loads could be expanded by providing valuable insight into the conditions under which a particular load goes unstable and what operational techniques delay onset of instability. A larger flight envelope increases throughput, an important factor for civil and military slung load operations.

In the design stage of a new helicopter or a new load configuration, a dynamic prediction tool would aid in verifying safety of flight prior to expensive wind tunnel or flight testing. A simulation with modular architecture, allowing the designer to assess different elements representing helicopter dynamics, load aerodynamics, and sling geometry, could allow for powerful parametric studies, showing the sensitivity of the complete design to variations in any of the basic elements of the system.

#### **D. SCOPE OF WORK PERFORMED**

This thesis describes the development and validation of a comprehensive dynamics and aerodynamics model for slung load simulation, obtained by integrating the NASA Ames Gen Hel UH-60A simulation with the equations of motion for slung load systems. Gen Hel is a proven component-type nonlinear Black Hawk helicopter model, using a blade-element implementation of the main rotor system. The main rotor dynamics in the model include rotor flapping, lagging, and rotor speed, incorporating lag dampers to enable the model to accurately predict possible rotor instability due to air resonance. The slung load dynamics represented single point suspensions using single or multi-cable slings, which could be elastic or inelastic, and includes load aerodynamics.

Helicopter main rotor downwash and load static aerodynamics were also modeled. The downwash model was constructed with airflow direction based on momentum theory assumptions and utilized empirical wake velocity data to determine the downwash field at the load. Static aerodynamics for the CONEX cargo container were obtained from wind tunnel tests at the Technion, Israel (Ref. [4]). The six-component static wind tunnel data provided a description of the forces and moments acting on the CONEX load at all angles of attack and sideslip angles which the load encountered throughout a range of conditions from hover to forward flight.

The collection and processing of simulation data was automated through the use of batch processing scripts, written to allow the user to set up a simulation run for a particular load configuration, control axis, and test airspeeds. Once started, the scripts directed the computer to run Gen Hel/SL cases, store the resulting time history output files, set up and execute the processing required to transform the time histories into the frequency domain, and determine system parameters from the appropriate frequency responses. In this way over 400 individual simulation test runs were completed and from these over 1600 system parameters were determined.

Validation of the Gen Hel/SL simulation was based on comparisons with flight test data. Frequency domain analysis was used to compare simulation and flight-derived frequency responses and key parameters of interest in the evaluation of system stability and handling qualities. The available flight test data from flight tests at NASA Ames were principally frequency sweeps over the range  $[0.05, 2]$  Hz, for several loads at hover and forward airspeeds (Ref. [5]). These flight tests represent over 100 separate data records, each containing the helicopter and load (where applicable) time history response to a control axis frequency sweep for a unique load configuration, control axis sweep, and trim airspeed. As with Gen Hel/SL simulation records, the flight data were processed through the use of automation scripts, yielding over 360 individual system parameters.

Results are given for the Gen Hel/SL simulation both with and without flight test loads. Handling Quality parameters, Stability Margins, and load pendulum motion roots for cases without load aerodynamic forces and moments, with drag-only aerodynamic estimations, and with static wind tunnel data were compared to flight test results. Additionally, the Gen Hel/SL model was used to predict system parameters for a 6K lb Block load.

These results illustrated the state-of-the-art in simulation modeling of helicopter/slung load dynamics and the accuracy in predicting key dynamic parameters of interest in handling quality studies. Of special interest was the use of flight and wind tunnel data on the load dynamics and aerodynamics in the model development.



## **II. NASA FLIGHT TESTING**

Flight testing with a utility helicopter and several different generic loads at NASA Ames Research Center commenced in April 1995. The slung load flight testing was divided into two parts, Phase I and Phase II.

The first portion of the Phase I tests, from May 1995 to October 1996, concentrated on procedure checkout and solid (non-aerodynamic) loads. Further Phase I test flights in July and August 1997 focused on the Container Express (CONEX) load, which is considered to have significant aerodynamics and was instrumented to provide motion data. The data from these tests have been accumulated in a NASA Ames database as described in Ref. [5].

Phase II flights were conducted from December 1998 through February 1999, using the instrumented 4K Block load. A brief description of the test equipment and conditions follows (a complete description of procedures, instrumentation, and equipment is found in Refs. [5] and [6]).

### **A. TEST AIRCRAFT DESCRIPTION**

The UH-60A Black Hawk is a utility twin-turbine, single main rotor helicopter capable of transporting cargo or up to 11 combat troops and weapons during day, night, visual and instrument meteorological conditions (see Figure 2). The aircraft has conventional wheel-type landing gear and four-bladed main and tail rotors. The helicopter is powered by two T700-GE-700 turboshaft engines each having an uninstalled rating of 1553 shaft horsepower at sea level, standard day static conditions. Installed dual engine power is transmission limited to 2828 shp. A cargo hook is mounted in the floor of the aircraft, and is gimbaled in roll. A moveable horizontal stabilator is located on the lower aft portion of the tail rotor pylon.

The test aircraft, serial number 82-23748 (ARMY 748), is a sixth year production Black Hawk which incorporates External Stores Support System fixed provisions and fairings,



reoriented production airspeed probes, and a modified production stabilator schedule.

Specifications of the UH-60A are shown in Table 1.

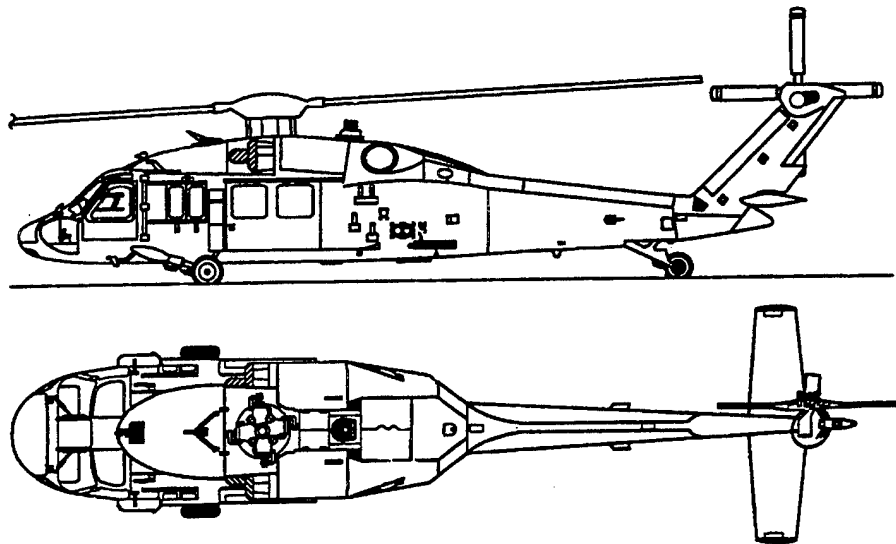


Figure 2. UH-60A General Configuration (after Ref. [7])

Operating Weights and Engine Power		
Empty Weight (lbs)	11,563	
Fuel Weight, Typical (lbs)	2,446	
Takeoff Weight, Typical (lbs)	14,609	
Maximum Takeoff (lbs)	20,250	
Maximum Takeoff Rating (shp)	3,086	
Maximum Useful Power (shp)	2,828	
Maximum Hook Capability (lbs)	8,000	
Rotor Parameters	Main Rotor	Tail Rotor
Radius (ft)	26.83	5.5
Chord (ft)	1.73	0.81
Solidity Ratio	0.082	0.188
Number of Blades	4	4
Rotor Rotational Speed (rad/sec)	27.02	124.54
Tip Speed (ft/sec)	725	685

Table 1. ARMY 748/UH-60A General Specifications (After Refs. [7] and [8])

Modifications made to the test aircraft under the airloads program and retained in the current aircraft test configuration includes the Aircraft Data Acquisition System (ADAS) comprised of several racks containing instrumentation and telemetry equipment and a flight data tape recorder. The ADAS includes air data sensors mounted on a nose boom, a low airspeed data system, and a total temperature sensor. The cargo hook, installed in the floor of the aircraft and gimbaled in roll, is instrumented with a strain gauge balance which measures total hook force. In addition, a video camera is mounted at the hook hatch and laser reflector assemblies are mounted on the landing gear stub wings.

A flow diagram of the instrumentation and data acquisition system on the helicopter and load is shown in Figure 3. The helicopter sensor signals are passed through filters and encoded in a Pulse Code Modulation (PCM) stream, which is recorded onboard and also transmitted to the ground telemetry station.

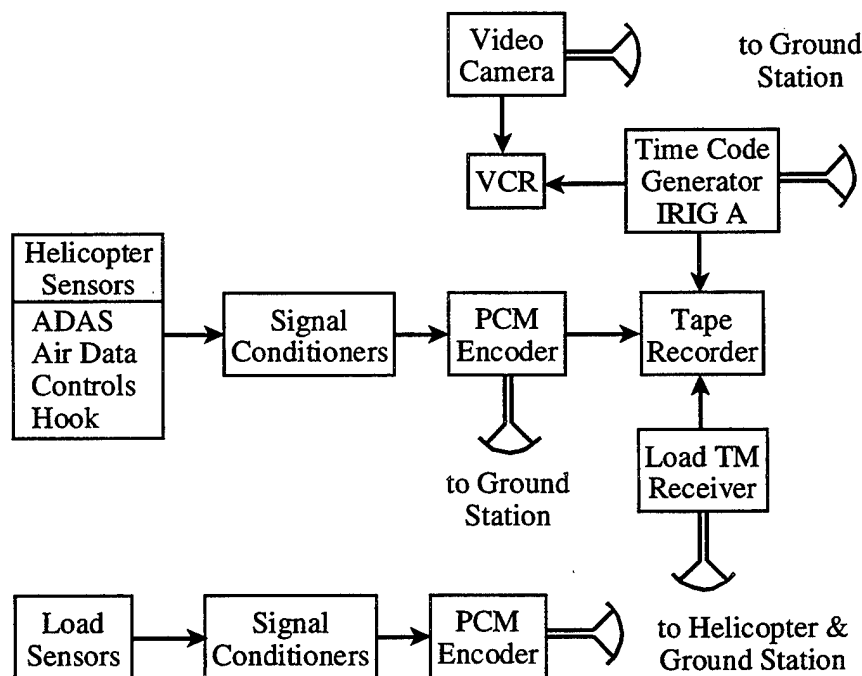


Figure 3. Helicopter and Load Data Acquisition System

A self-contained portable load instrumentation package contains 3-axis accelerometers and rate gyros, pitch and roll inclinometers, signal conditioning equipment, a PCM encoder, and battery. The transmitting antenna can be mounted on the cover of the package, and a fluxgate compass is installed on the load at the end of an aluminum boom in order to minimize compass errors due to magnetic influence of the load. Similar to the helicopter data signals, the load sensor signals are filtered, encoded, and transmitted. The load instrumentation package was provided to NASA Ames by the Israeli Air Force under the US/Israel MOA (Ref. [3]) and is described in detail in Ref. [5].

## **B. FLIGHT TEST LOADS**

Data for the validation study came from two test loads illustrated in Table 2. The sling used was a standard 4-legged military sling rated at 10 K lbs capacity. The loads were fastened to the sling at the four upper corners. The 4K lbs steel Block had the instrumentation package mounted on the top surface and the magnetic compass mounted on an aluminum boom extending from the side. This was a high density test load with negligible aerodynamic forces and moments over the power-limited speed range of the helicopter, and thus for this load aerodynamic forces and moments were not a factor in validating the simulation.

The second load was an 8 ft x 6 ft x 6 ft CONEX which possessed significant aerodynamic effects even in hover where rotor downwash results in a steady yaw rotation. It was limited to 60 kts in military operations (Ref. [9]), well below the 140 kt UH-60A power-limited speed. Mass-inertia-geometry parameter values for the sling and loads are included in Table 2.

The principal load dynamics affecting aircraft stability margins and handling qualities are the load pendulum modes. Since the configurations tested in this analysis were multiple sling suspensions with relatively inelastic legs thereby removing load pitch and roll degrees of freedom, the load motion could be estimated as a compound pendulum.

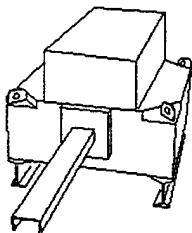
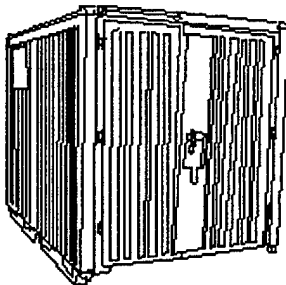
<b>4K Block</b>		<b>4K CONEX</b>		
				
Weight:	3895 lbs	Weight:	4105 lbs	
Density:	365 lb/ft <sup>3</sup>	Density:	12.5 lb/ft <sup>3</sup>	
Dimensions:	2.64 x 2.64 x 1.22 ft	Dimensions:	8.48 x 6.41 x 6.11 ft	
<b>Moments of Inertia:</b>	<b>4K Block</b>	<b>4K CONEX</b>	<b>ARMY 748</b>	
$I_{xx}$	103	1876	5629	lb-ft-s <sup>2</sup>
$I_{yy}$	103	1482	40000	
$I_{zz}$	174	1377	37200	
Aircraft Center of Gravity to Hook Coordinates:			0.98, 0, 4.3 ft (nominal)	
<b>Sling Parameters:</b>				
Initial Cable Length:	$l_0$	15.83	Ft	
Spring Constant:	$K$	9645	lbs/ft	
Damping Constant:	$C$	22	lbs/ft/sec	

Table 2. Load Configuration Geometry and System Mass Parameters

One and two degree of freedom (DOF) compound pendulum models were used to estimate the load pendulum natural frequencies in order to determine if interaction between the load and the main rotor regressive lead-lag mode was to be expected. Table 3 shows the compound pendulum natural frequencies computed by the two models according to the derivation found in Appendix A. The frequencies for both the Block and CONEX loads were no higher than 2.1 rad/sec. The unstable regressive lead-lag frequency for the Black Hawk helicopter is about 0.25 times the rotor rotational velocity, in this case about 6.8 rad/sec, and so the estimated pendulum mode for the two loads was low enough to prevent possible air resonance. If the sling length were shortened by 10 ft, the highest natural frequency found through the one and two DOF approximation reached nearly 3.0 rad/sec, still well below the region of concern for air resonance.

Load	Axis	Two DOF		One DOF
		$\omega_n$ (rad/sec), fundamental	$\omega_n$ (rad/sec), second	$\omega_n$ (rad/sec)
4K Block	Lateral	1.152	2.090	1.397
	Longitudinal	0.6256	1.445	
4K CONEX	Lateral	1.114	2.052	1.301
	Longitudinal	0.6401	1.348	

Table 3. Load Pendulum Natural Frequency Estimates

The flight tests conducted to date focused on the lateral and longitudinal response of the helicopter with the various loads at hover to 80 kts forward airspeed. Flights without a slung load were conducted in order to provide a baseline condition for comparison of the helicopter computer model. Table 4 shows the airspeeds at which each load was flown.

Load	Airspeed (kts)					
	Hover	30	50	60	70	80
No Load	X	X	X			X
4K Block	X	X	X			X
4K CONEX	X	X	X	X	X	
Note: Load Instrumentation Package not installed for 80 kt 4K Block case.						

Table 4. Flight Test Matrix

### C. FLIGHT TEST DATA ANALYSIS

The flight tests were designed to provide high quality helicopter and load motion response data due to pilot-generated frequency sweeps, primarily in the lateral and longitudinal axes. During the flight testing sweeps in the yaw and the collective axes, along with control doublets and steps, were performed and recorded, however the basic helicopter response was primarily affected by the addition of the external load in the lateral and longitudinal axes.

The yaw axis was not of concern for Handling Qualities (HQ) because in the single hook configuration of the UH-60A the hook did not transmit a yaw moment from the sling to the aircraft. Instead the sling "wound up" as the load yawed, and then reversed its spin direction when the number of turns on the sling and the accumulated torsional moment at the hook was sufficient to counter the load aerodynamic driving force and rotational inertia. This occurred at a frequency well below HQ frequencies of interest, between 1 to 10 radians per second. The wind tunnel data collected for the CONEX load aerodynamic model was static, therefore it did not include any additional aerodynamic effects from the dynamic motion of the load in yaw.

Since the main thrust of the helicopter/slung load testing was to explore handling quality and stability margin, frequency domain analysis of system response to control input frequency sweeps was used. For an overview of AFDD/NASA frequency domain analysis techniques see Refs. [10] and [11].

The time history data from flight control input sweeps was converted into frequency responses with CIPHER<sup>®</sup> which used advanced spectral analysis with the Chirp-Z transform and composite windowing techniques. CIPHER<sup>®</sup> contained extensive analysis modules tailored to extract flight parameters and assist in the comparison of the flight data to simulation results, including: 1) Handling Qualities and Stability Margin analysis; 2) transfer-function identification through the NAVFIT routine; 3) frequency response arithmetic functions; and 4) plotting and database management.

## **1. Handling Quality**

Handling Quality parameters are a measure of the response of the helicopter to a control axis input. HQ is determined from the Bode Plot of the helicopter attitude angle to the control input (i.e.,  $\phi/\delta_a$  for the lateral axis and  $\theta/\delta_b$  for the longitudinal) and is given as two values, bandwidth and phase delay. As shown in Figure 4 for a rate response-type helicopter without

attitude command or attitude-hold engaged, bandwidth,  $\omega_{bw}$ , is the lower of the frequencies given by the crossing of  $-135^\circ$  phase or 6dB magnitude greater than the magnitude associated with the  $-180^\circ$  phase crossing. Phase delay,  $\tau_{pd}$ , is determined by the slope of the phase plot as it crosses  $-180^\circ$  by a linear least squares fit if the phase response is not linear between the crossing frequency and twice the crossing frequency. (Ref. [1])

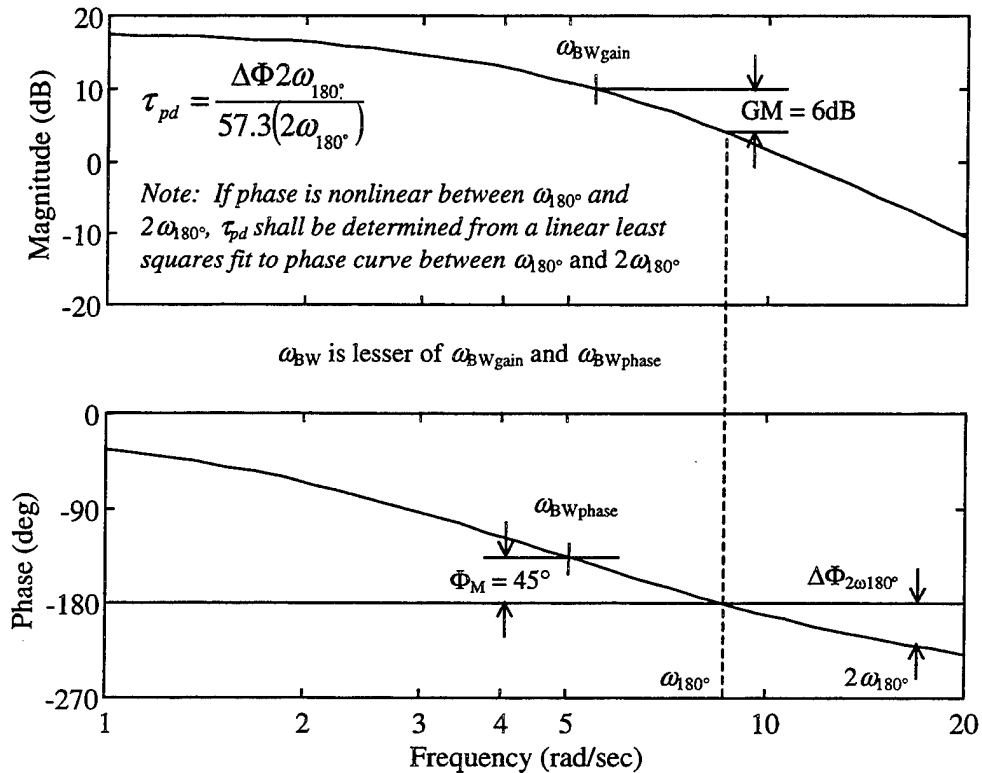


Figure 4. Handling Quality Determination (After Ref. [1])

In this analysis of the flight data, a coherence-weighted linear least squares fit was performed between frequencies bounding the linear region of the phase response as it crossed  $-180$  deg. Although this method is different than the definition given in ADS-33D, it was chosen due to poor coherence and highly non-linear characteristics found in most of the flight phase responses at the higher frequencies near  $2\omega_{180^\circ}$ .

As  $\omega_{bw}$  decreases the perceived performance of the helicopter also decreases. The tolerance for decreased bandwidth is less as  $\tau_{pd}$  increases. Therefore a good design would have a short time delay and a large bandwidth, allowing for optimum response to a pilot input.

Past helicopter HQ analysis has developed criteria for acceptable performance in the lateral and longitudinal axes for specific aircraft mission and visual conditions. In the absence of specific criteria for the UH-60A with an external load, HQ values and pilot rating level boundaries for the cargo helicopter given in ADS-33D were used for comparisons.

## 2. Stability Margin

Stability Margin (SM) is a measure of the Phase and Gain Margin present in the control axis frequency response at the associated 0 dB or  $-180^\circ$  crossing. The presence of a load affects the helicopter's frequency response in such a way that the SM may be decreased, since the addition of the load's pendulum mode into the overall system response may cause a dip below the 0 dB line at a lower frequency than the basic helicopter's (no load) response. Stability Margin is computed from the bode plot as shown below in Figure 5.

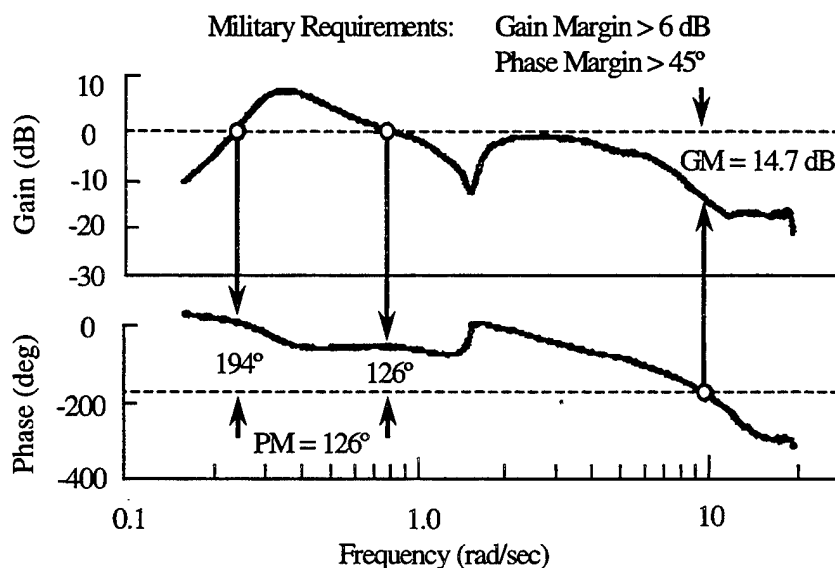


Figure 5. Stability Margin Determination



The frequency response used for SM determination is the measure of the Stability Augmentation System (SAS) output to the total control signal in the helicopter axes that have active SAS loops. Flight test measurement of this response was determined in two different ways, designated as the *Direct Method* and the *Indirect Method*. The difference between the two methods is illustrated in the simplified model of the helicopter control system shown in Figure 6, where the Direct Method is a measurement of  $f_{SAS}(s)/e(s)$  combined with a simple gain to represent the linkage dynamics and the Indirect Method is derived from the measurement of  $e(s)/r(s)$  according to the formula (Ref. [5]):

$$\frac{f(s)}{e(s)} = \left( \frac{e(s)}{r(s)} \right)^{-1} - 1 \quad (1)$$

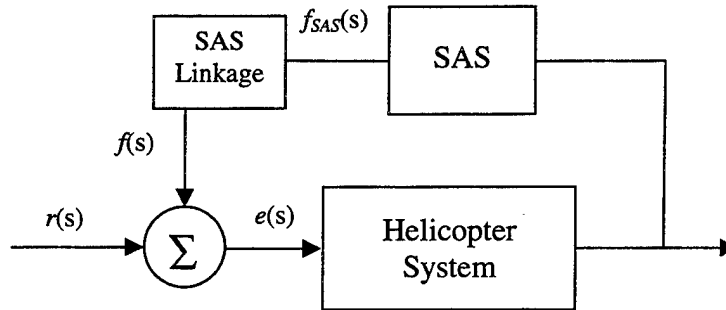


Figure 6. Simplified Helicopter Control Model

Analysis of the flight test data from the NASA Ames Slung-Load testing showed that the Indirect Method maintained better coherence in the frequency range near load motion. The small differences between the two methods are due to dynamic effects of the summation junction found on the aircraft, which generally acts as a time delay in the higher frequencies. Thus the differences in Phase Margin, determined at lower frequency, shows little change while the Gain Margin, which relies on the  $-180^\circ$  crossing, exhibits greater variation.

### 3. Load Pendulum Motion

While not a direct measure of the helicopter's performance in slung load operations, load pendulum motion parameters were computed to verify a complete understanding of the forces and moments acting upon the two-body system. An undamped or under-damped load pendulum mode indicates a stability problem and could result in damage to the helicopter or loss of load. Additionally, a swinging load can add danger to ground personnel, as the load motion during the load pick-up or drop-off could present a danger.

As done in previous work (Ref. [5]), the load pendulum mode was determined through the use of NAVFIT, which computes the damping ratio and the natural frequency of a second order transfer function that matches closely the load response. The fit was performed over a small range of frequencies around the load root, and thus approximated the load motion without the overall dynamic effect of the helicopter system. A sample NAVFIT plot is shown in Figure 7.

Hover Lateral Pendulum Mode:  $\zeta = 0.166$ ,  $\omega_p = 1.53$  rad/sec

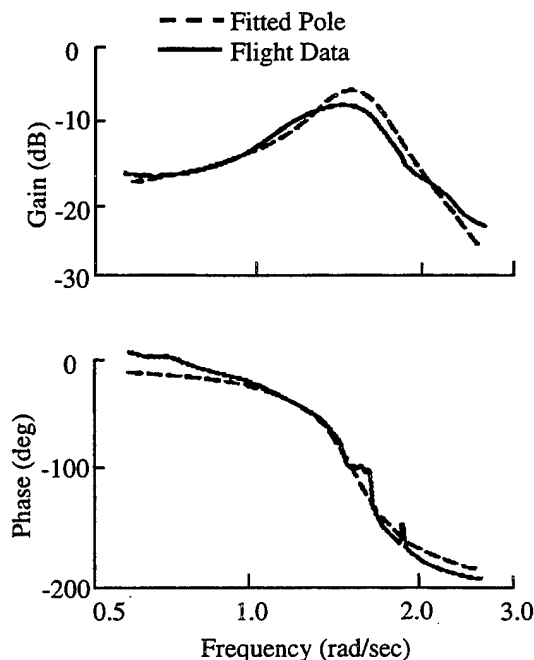


Figure 7. Load Pendulum Mode Determination With NAVFIT



### **III. SLUNG LOAD SIMULATION**

Efforts at NASA Ames to model the helicopter/slung load system led to the development by Mr. Luigi Cicolani of a linearized, 12 degree-of-freedom model according to the Equations of Motion detailed in Ref. [12]. This FORTRAN routine, called `SL_DRIVER`, is capable of using the stability derivative models of several different airframes, including a CH-47D, UH-60, and NASA's Enhanced Stability Derivatives (ESD) aircraft model. Additionally, the static aerodynamics for the 8 x 8 x 20 ft cargo container (MILVAN) were used with the CH-47D model.

Based on the success of the CH-47D simulation, the slung load dynamics of `SL_DRIVER` were installed as a module in the NASA Ames UH-60 Gen Hel model, instead of continuing to develop `SL_DRIVER`'s internal linear aircraft models. Additionally, wind tunnel derived static aerodynamics for the CONEX was incorporated and a rotor downwash model was included to simulate the relative wind experienced by the load in low speed level flight.

#### **A. GEN HEL ADVANCED ROTOR/HELICOPTER MODEL**

The Sikorsky-Ames Gen Hel non-linear mathematical model of the UH-60A Black Hawk helicopter was developed under contract for the US Army and NASA by Sikorsky Aircraft. The model, as described in detail in Ref. [13], was based upon the Sikorsky General Helicopter Flight Dynamics Simulation, and was intended to provide an engineering simulation suitable for performance and handling quality evaluation. The real-time version of the program has been validated and used for pilot-in-the-loop VMS simulation (Ref. [14]), however Gen Hel Version 6.0 used in this study was intended for non-real-time use.

The model represents the single main rotor helicopter by a six degree-of-freedom rigid body, including rotor blade flapping, lagging, air mass, and hub rotational degrees of freedom. The simulation is comprised of program modules representing the major helicopter subsystems, with a detailed interface of the physical quantities such as forces, moments, attitudes, and velocities shared between the modules. The modular nature of Gen Hel allows for individual modification or interchange of any of the elements, a characteristic that makes the Gen Hel program suited for the slung load study. The major components of Gen Hel are given in Figure 8.

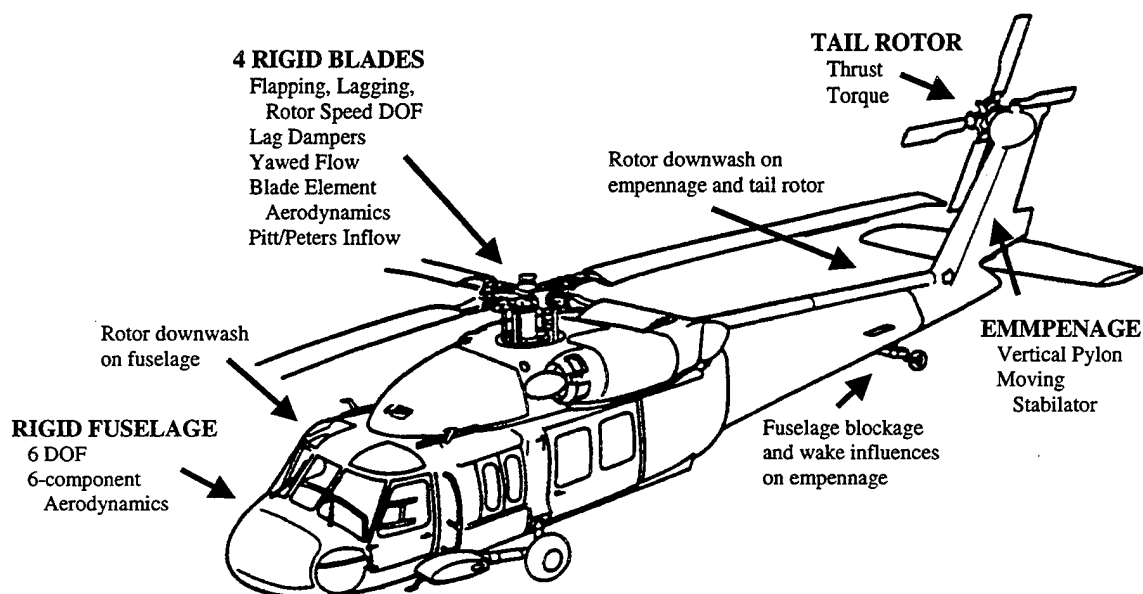


Figure 8. NASA Ames Gen Hel UH-60A Components (After Ref. [15])

Blade-element theory is used for the main rotor: total rotor forces and moments are computed as the summation of aerodynamic, inertial, and gravitational forces on each of the 5 elements on each blade. Dynamic inflow at each rotor segment is computed with the Pitt/Peters inflow correction (Ref. [16]), which is based on unsteady actuator-disk theory. The addition of the Pitt/Peters correction is one of the refinements made at NASA Ames, along with corrections

and expansions to the rotor blade equations of motion and the addition of a thermodynamic-cycle component model of the T700 engines and the rotor drive train.

Linearized Bailey theory (Ref. [17]) is used in the tail rotor thrust computation. As shown in Fig. 8, the model accounts for rotor downwash effects on the empennage, fuselage, and tail rotor. This is done through the use of empirical flight test and wind tunnel test data or from analysis-oriented simulations, which are incorporated in the program as look-up tables.

The flight control system modeled in Gen Hel matches the physical setup of the UH-60A Black Hawk flight control system, allowing the computer simulation output channels to correspond directly to those measured on the test aircraft. In this way a complete comparison on flight and simulation can be conducted, not only of the overall aircraft state variables, but also of various control feedback loop responses.

An important aspect of the flight control model in Gen Hel is that the SAS input summation junction (as shown in Fig. 5) is modeled as a simple gain, without the linkage dynamics noted earlier. Because of this the Gen Hel-derived Stability Margin frequency responses from either the Direct or Indirect Method are identical. For ease of analysis, the Direct Method was computed during Gen Hel/SL data processing.

## **B. SLUNG LOAD DYNAMICS**

The two body equations of motion for general multi-cable slings suspended from a single point were implemented as given in Ref. [12]. Details of the slung load equations of motion are shown in Appendix A. The generic configuration is shown in Figure 9 and represents slings with 3 or more legs attached at lift points on the load such that at least 3 sling leg directions are independent. The bodies are assumed rigid and the sling legs are elastic or inelastic. The hook-sling attachment is modeled as transmitting forces but not moments.

The parameters required for these equations are the masses and inertia matrices of the two bodies, the helicopter-c.g.-to-hook coordinates, the load-c.g.-to-lift-point coordinates, and the unloaded cable lengths and cable stretching parameters. Parameter values for the test configurations are listed in Table 2.

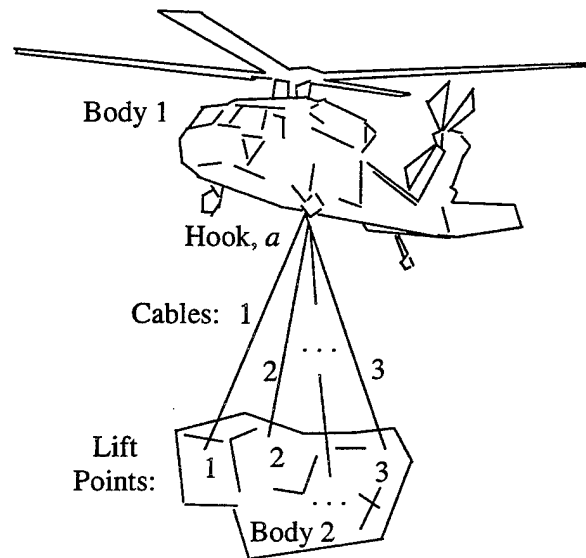


Figure 9. General Multi-Cable Sling Configuration

Sling stretching is conventionally modeled as a lightly damped spring which supports only tension. Parameter values for this model were identified in dynamic tests (Ref. [5]). However, moving-base piloted simulation studies at Ames found that the elastic cable model results in excessive hook force excursions applied to the aircraft and sensed by the pilot, and was unrealistic. Consequently, only results obtained from Gen Hel/SL with inelastic cable dynamics are given in this work although the simulation has provisions for elastic or inelastic slings. This suffices for present purposes since cable stretching dynamics have not been observed in the frequency range of this study. However, they have been implicated in incidents involving vertical bounce dynamics and would be essential in studying higher frequency load-airframe-rotor interactions such as air resonance.

Simulations of isolated helicopters are normally initialized in static equilibrium using a gradient search computational procedure. The load-sling combination was readily integrated into this scheme by computing the hook force at each iteration from the force and moment balance equations of the load-sling subsystem. For inelastic slings, load moment balance was used to compute load attitude after which the inertial components of the hook force were computed from load force balance. Variations in load aerodynamics with attitude were treated by iteration of small nonlinear terms in the load-sling equations starting from an initial attitude estimate. If the sling was elastic, then the effects of stretching on load attitude would be included in the iteration. The effects of downwash on load aerodynamics were included in the "outer loop" iteration of the helicopter equations.

### **C. MAIN ROTOR WAKE MODEL**

In considering the sources of aerodynamic forces and moments on the load and the resulting impact on the helicopter's motion, the characteristics of the rotor wake must be taken into account. In a no-wind hover, the wake proceeds downward from the rotor disk as a helix oriented along an axis perpendicular to the Tip Path Plane (TPP), contracting to a minimum diameter by approximately 1.5 times the rotor radius distance. When wind is introduced or if the helicopter is in forward or sideward level flight, the wake veers to the down wind side as it proceeds away from the disk.

#### **1. Wake Geometry**

Momentum Theory is useful in describing the characteristics of the wake, general geometry, and approximate velocities. From this theory and a few assumptions described below, the wake was modeled and described in its own set of moving coordinate axes as shown in Figure 10. The near-wake velocity at the rotor disk,  $V'$ , was determined from the rotor inflow velocity



and the hub translational velocities, both of which are provided by Gen Hel's rotor blade subroutine. The far-wake velocity,  $V''$ , computed from the same values, however the axial inflow velocity component was doubled as the rotor wake contracts.

The Euler angles describing the rotations about the inertial axes to describe the wake coordinates are given by the relations:

$$\begin{aligned}\psi_w &= 0 \\ \theta_w &= \text{atan}\left(\frac{u_N''}{w_N''}\right) \\ \phi_w &= \text{asin}\left(\frac{-v_N''}{V}\right)\end{aligned}\tag{2}$$

where  $u_n''$ ,  $v_n''$ , and  $w_n''$  are the inertial axes components of  $V''$ . All of the coordinate transformations used in the Gen Hel/SL program are described in Appendix A.

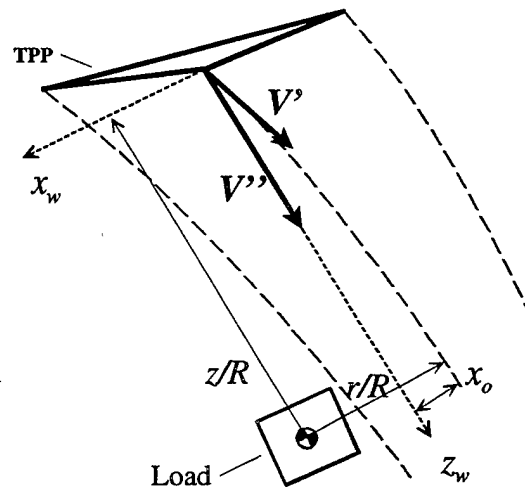


Figure 10. Wake Coordinate System

Since the shape of the wake is curved as it follows the arc from the rotor disk to the far-field wake, the center of the wake is offset from the position that would be obtained simply by following the  $x_w$  direction from the center of the rotor hub. The  $x$ -axis offset is shown in Figure

10 as  $x_o$ , and a similar offset occurs in the  $y$ -axis direction due to sideward flight. Following the derivation presented in Ref. [18], the offsets were calculated through the use of the following simplifications:

- The wake is assumed to be fully contracted by 1.5 times the rotor radius downstream from the rotor disk.
- The time to travel this distance,  $\tau$ , can be taken from the average of the two wake velocities,  $V'$  and  $V''$ .
- Fuselage interaction and interference in the wake is not considered.

The wake center offsets  $x_o$  and  $y_o$  were computed as:

$$\begin{aligned} x_o &= \frac{u'_w}{2} \tau & \tau &\approx \frac{1.5R}{\left(\frac{V' + V''}{2}\right)} \\ y_o &= \frac{v'_w}{2} \tau \end{aligned} \quad (3)$$

In addition to the wake core offset, an elliptical correction to the wake boundary was applied. As flight speed increases, the cross section of the wake becomes elliptical, until at high speed the wake is nearly flat, with a semi-span equal to the rotor blade radius. To account for this effect, the non-dimensional horizontal  $r/R$  distance from the center of gravity of the load was corrected to account for elliptical contraction (Ref. [15]). The elliptical correction acts to remove the load from the influence of the main rotor wake at a lower forward airspeed than if the correction was not applied.

## 2. Downwash Velocity

With the wake geometry computed with variables available in Gen Hel's module interface structure, and the position of the load's center of mass identified relative to the center of the rotor blade hub, the magnitude of the main rotor downwash acting on the load was estimated

from empirical rotor wake data. Using test results from Ref. [19] for wake velocity in the axial direction, combined with an empirical relationship between axial and tangential wake flow (Ref. [20]), values for the wake velocity magnitude as shown in Figure 11 were developed. In the figure the axial velocity component is given as the ratio of dynamic pressure to the rotor Disc Loading,  $q/DL$ , which allows the magnitude of the velocity to be scaled for ambient conditions as well as by helicopter/slung load gross weight variation. The tangential component is shown as a fraction of the axial component. In the center of the wake, the ratio approaches zero (no tangential velocity) to a maximum of 10% in the outer portion of the wake.

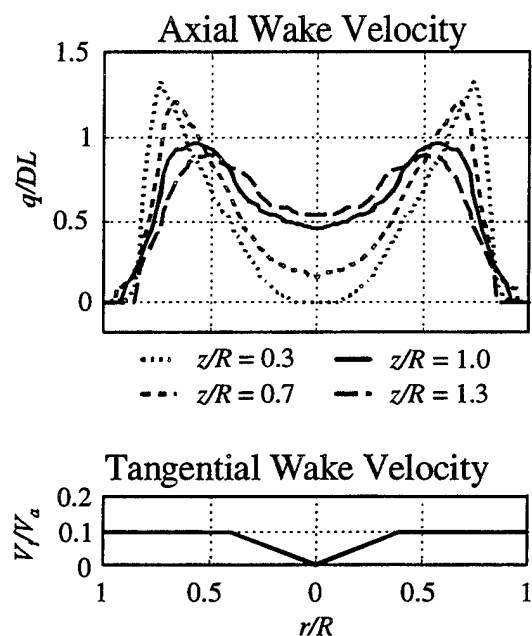


Figure 11. Wake Velocity Components

#### D. LOAD AERODYNAMIC FORCES AND MOMENTS

The available measurements of load aerodynamics come principally from studies of the 8 x 8 x 20 ft MILVAN cargo container and a few other loads made in the early 1970's in support of the heavy lift helicopter development (Refs. [21], [22], [23], [24], [25]). Currently, wind tunnel

studies have been conducted on the HUMVEE vehicle in support of V-22 Osprey development (Ref. [26]) and on the CONEX for this project. Wind tunnel measurements have usually been limited to the load's static aerodynamics (steady state variations with air velocity direction), and to studies of the critical airspeed at which an aerodynamically active load becomes unstable. A comprehensive model structure is not available for loads, but is expected to include the effects of load angular rates and unsteady aerodynamic phenomenon.

As with all slung loads, the CONEX was observed in the flight tests to adopt a steady trail angle in proportion to drag. The CONEX also exhibited significant yaw rates which were as much as 45 deg/sec in hover due to swirl in the rotor downwash, and which increased past 100 deg/sec for airspeeds above 50 kts. The load yaw rate was steady if a swivel at the cargo hook was used, and periodic with sling wind-up and unwinding if no swivel was used.

#### 1. Drag Force Only Estimation

The simulation included options for drag-only load aerodynamics and the CONEX static aerodynamics. In the drag-only estimation, load drag was represented by the parameter,  $D/q$ , independent of airspeed. The load body-axes aerodynamics are:

$$\mathbf{FA}_{22} = (D/q) 0.5 \rho \mathbf{Va}_2 \quad \mathbf{MA}_{22} = (0, 0, 0) \quad (4)$$

where  $\mathbf{FA}_{22}$ ,  $\mathbf{MA}_{22}$  are the load aerodynamic force and c.g. moment vectors,  $\mathbf{Va}_2$  is the airspeed at the load center of gravity. For many loads a single value of  $D/q$  independent of velocity direction suffices. For the CONEX load the value of  $D/q$  varies from 42 to 88 ft<sup>2</sup> depending on direction. For comparison, drag for the more elongated MILVAN varies from 60 to 210 ft<sup>2</sup> depending on orientation.

## 2. CONEX Static Aerodynamics

Wind tunnel tests were conducted at the Technion in Israel to measure the static aerodynamics of the CONEX (Ref. [4]). A 5.7% scale model was manufactured (5.8 x 4.5 x 4.5 in), including the corrugations of the CONEX wall and the skids. The model was mounted on a sting balance that measured all six aerodynamic force and moment components. Inside the model there was a mechanism to change the pitch and roll angles (shown in Figure 12) over a range of  $\pm 25^\circ$  relative to the balance.

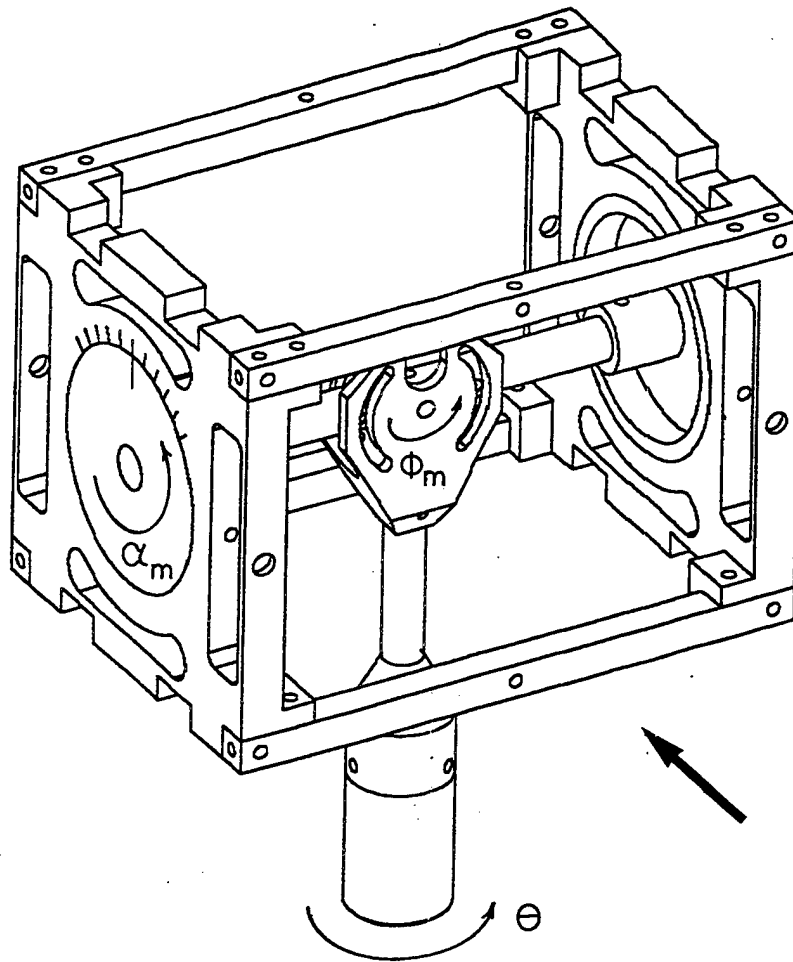


Figure 12. CONEX Wind Tunnel Model Internal Mechanism (From Ref. [4])

The tests were carried out at the Technion's open-circuit fan-driven low-speed wind tunnel, which has a 3 ft x 3 ft by 10 ft long test section and can reach speeds of 58 kts (Figure 13). The tunnel has good uniformity of flow within 1% over the cross-section except very near the tunnel walls. There were various mechanisms for mounting the model and sting. For the CONEX tests, the model and sting were mounted on the *banana* arm (Figure 14) which is a circular arch mounted in the middle of the test section floor to a rotating plate and supported by a bearing in the test section ceiling. The sting balance was connected to an extension arm attached to the arch and the extension arm can be positioned in intervals of  $2.5^\circ$  from vertical to  $10^\circ$  above the horizontal. This arrangement maintained the model in the center of the tunnel cross-section while allowing a range in model pitch over  $[-115, 25]$  deg and a  $360^\circ$  of yaw.

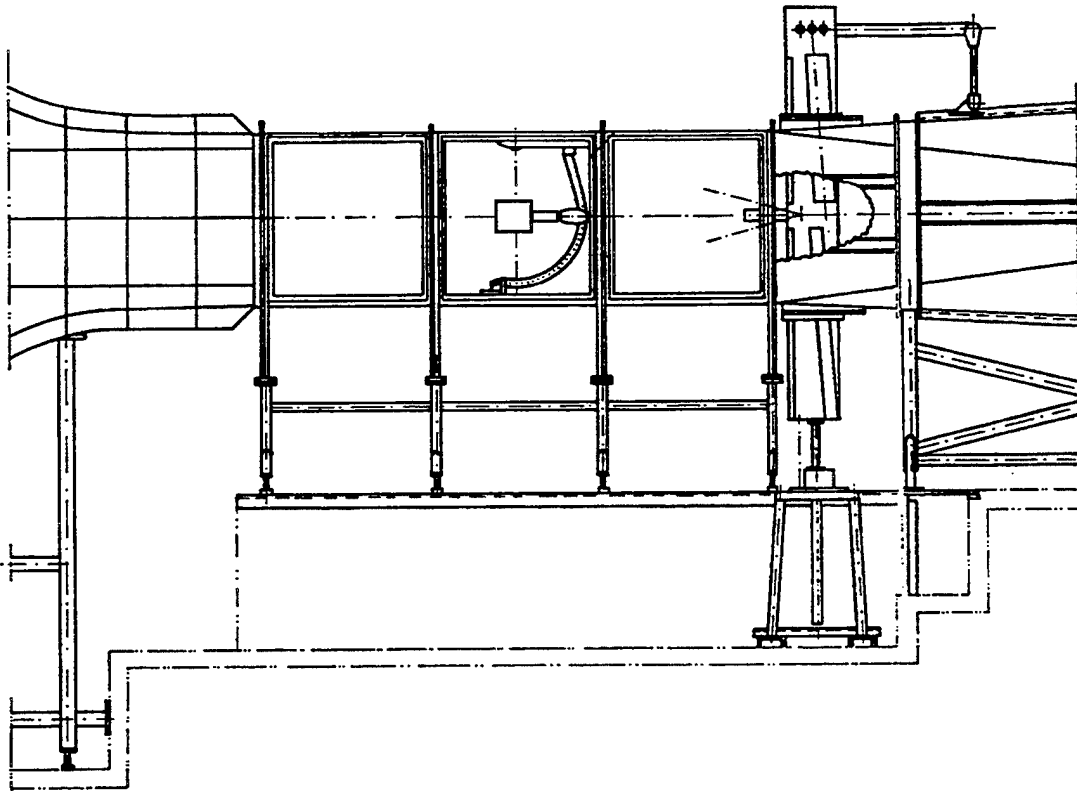


Figure 13. Technion Low-Speed Wind Tunnel

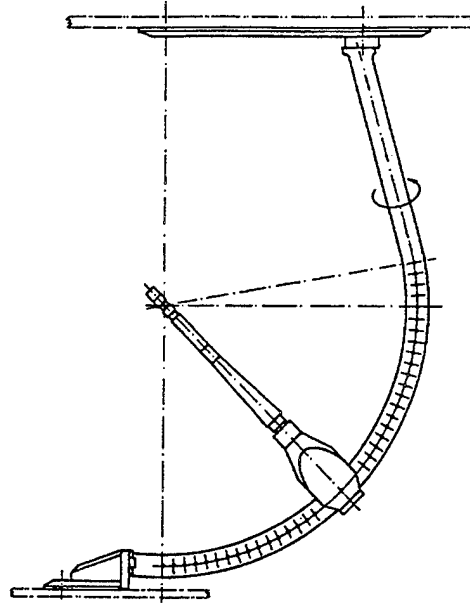


Figure 14. Wind Tunnel Banana Arm (From Ref. [4])

The tests were conducted at 39 kts tunnel speed. Tests at several flow speeds indicated negligible variations of measured coefficients with tunnel speed. Runs were performed by rotating the banana arm over  $\pm 180^\circ$  in yaw about the tunnel vertical axis at 2.5 deg/sec with fixed model angles relative to the sting and fixed mounting angle of the extension arm on the banana arm. During the rotation, measurements of the flow speed, arch angle relative to the flow, and the six aerodynamic components were made at a data rate of 5 KHz. The data were normally averaged over 0.1 sec intervals (yaw intervals of  $0.25^\circ$ ). Each run included more than 700 points and more than 30 runs were made for the CONEX data. Wind axes results were computed for grids of  $\alpha, \beta$  every 5 deg. The  $\alpha, \beta$  values for each measurement point were computed and the measurements assigned to *bins* corresponding to each of the grid points. Statistics were computed for each bin, and symmetry rules could be applied between appropriate bins.

### 3. Simulation Static Aerodynamic Model

Tables of the static aerodynamic parameters were derived from the tunnel measurements for the angle of attack and sideslip domain  $\alpha, \beta = [-90, 90] \times [0, 90]$  deg. Extension to negative sideslip uses symmetry properties (drag and lift are symmetric, side force and yaw moment are anti-symmetric). Small modifications of the tunnel data were made to impose some symmetry properties; that is, side force and yaw moments pass through zero at  $\beta = 0, 90^\circ$  and drag is fixed for  $\beta = 90$  deg. An apparent fixed bias was removed from the lift function. No modifications were made to impose expected symmetries in AOA about  $0^\circ$  pending further tests for systematic tunnel errors. Tunnel measurements were not made for  $\alpha > 30^\circ$  as excursions above  $30^\circ$  are unlikely in flight. Nevertheless, the data was extended into this region by linear extrapolation for the simulation model. The results are shown in Figure 15 where the aerodynamics are plotted versus sideslip (or AOA) for fixed values of AOA (or sideslip) every 10 deg. Drag is the largest force; reaching a minimum at  $90^\circ$  sideslip where the CONEX has the minimum frontal area and looks identical to the axial flow independent of pitch, and tending to increase with AOA owing to the skids on the bottom of the CONEX which trap air. Side force is positive at all positive sideslip with similar variations versus sideslip for all AOA. For small sideslip angles lift is approximately anti-symmetric in AOA about zero and reaches peak values around 15 deg. This behavior is repeated in the vicinity of  $\alpha = -90$  deg. For sideslip angles 20 deg, lift is small everywhere. Yaw moment has similar behavior at all AOA and is stable at  $\beta = 0, 90$  deg.

### E. GEN HEL/SL SIMULATION

The combined Gen Hel and Slung Load dynamics simulation structure is shown in Figure 16. The integration of the slung load elements into the Gen Hel program was done with a minimum number of changes to the Gen Hel software. Separate variables and common files were



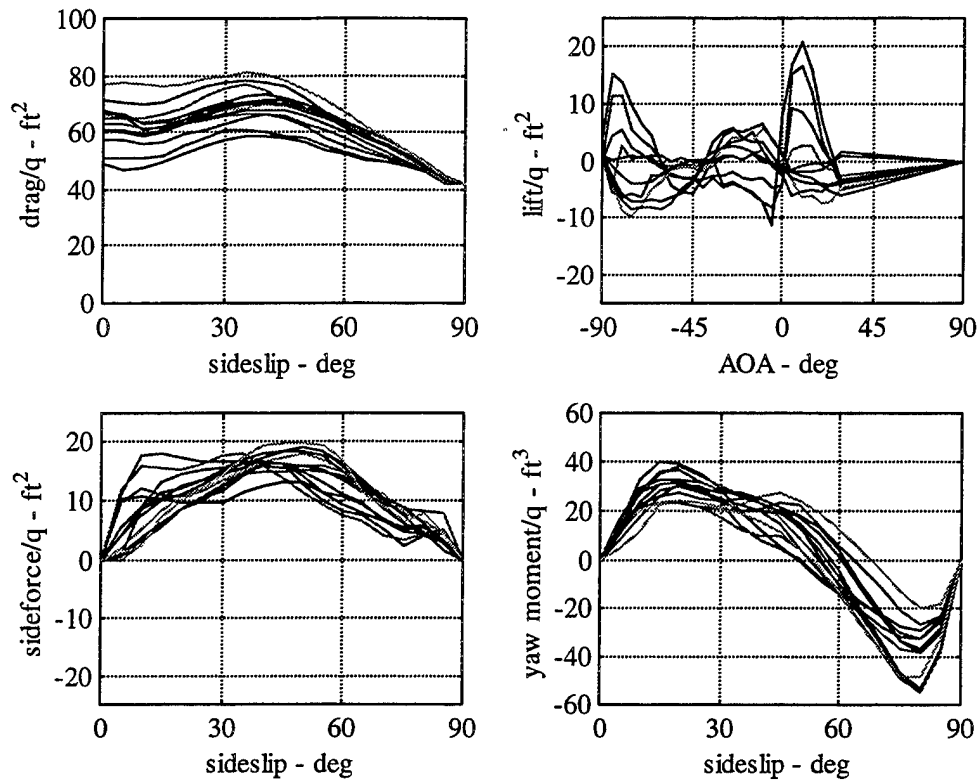


Figure 15. CONEX Static Aerodynamic Coefficients

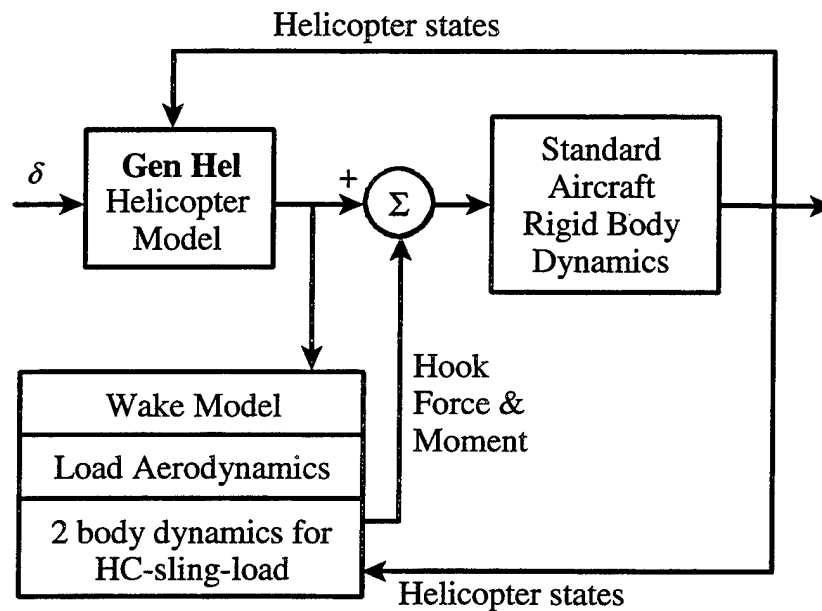


Figure 16. Gen Hel/SL Simulation Architecture

used wherever possible. The Gen Hel modules were able to provide the slung load subroutines with all necessary data, including the aerodynamic forces and moments acting on the total helicopter system and various main rotor parameters used for rotor wake computation.

The Gen Hel/SL model was run on a UNIX workstation and modified to allow output time histories to be entered directly into the CIPHER<sup>®</sup> using GETDATA UNC3 files. The program structure and FORTRAN source code used in the incorporation of the SL\_DRIVER slung load dynamics into Gen Hel are given in Appendix B.

### **1. Computer-Generated Control Sweep Input**

The Pilot Input mode of Gen Hel was modified so that a binary input file containing control information could direct the aircraft after initial trim position was established. Selected state variables and control system values were then sent to a time history output file. In this way either actual test flight control sweeps (pilot-generated) or computer-generated frequency sweeps could be used as the basis for the analysis. A sample control input sweep created by the program `makesweep` is shown in Figure 17. White noise is added to the on- and off-axis channels.

### **2. Control Feedback Loop**

The difficulty in using frequency sweeps in an unpiloted simulation model was maintaining airspeed and attitude close to initial trim values. Sweep durations of 100 seconds were used, and so additional control was provided as a three channel low-gain rate and attitude feedback loops as used by Mansur, et al. (Ref. [27]). The multi-input/single-output spectral analysis (MISOSA) routine in CIPHER<sup>®</sup> removed off-axis to on-axis correlation and the resulting frequency response represented the relationship of the output to the input with the off-axis inputs removed. The lateral axis sweep shown in Figure 17 is with the feedback loop active for hover with the CONEX load. The off-axis inputs in this case are minimal, while the effects of the feedback loop can be seen in the on-axis time history.

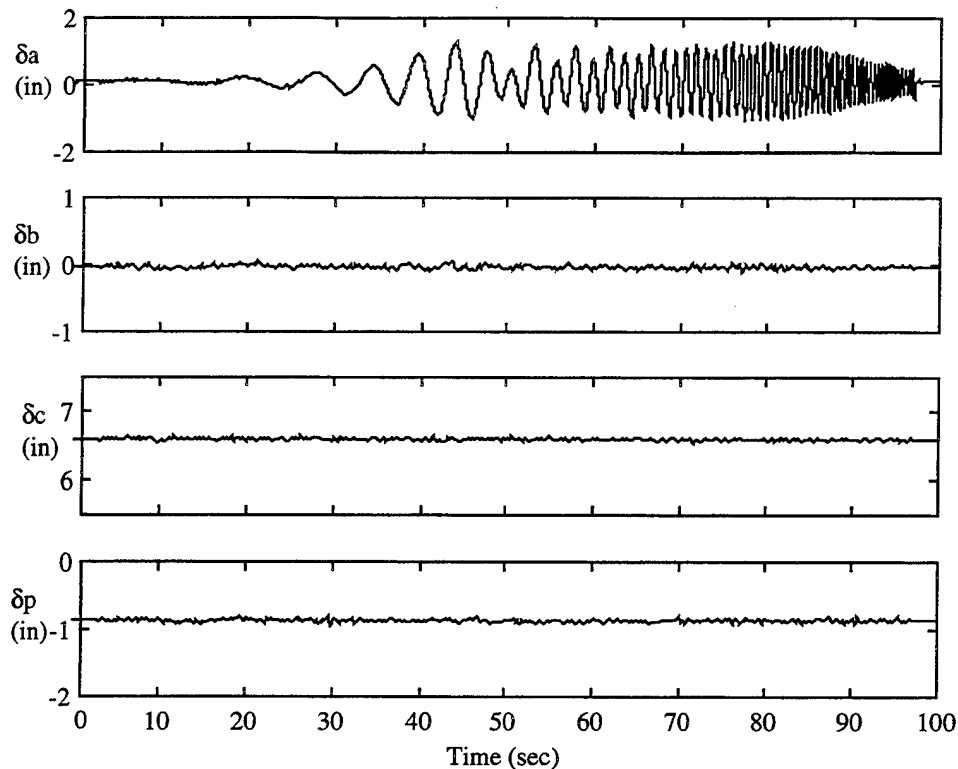


Figure 17. Computer Generated Frequency Sweep

### 3. Data Collection and Processing Automation

In order to run the complete Gen Hel/SL simulation for multiple load configurations, helicopter fuel weights, airspeeds, and control input axes, or to rapidly analyze flight data, a combination of C-Shell and Expect scripts were developed. The C-Shell scripts controlled the non-real-time execution of Gen Hel/SL, the conversion of the output data to UNC3 file format, and setting up and running of CIPHER<sup>®</sup> cases; all functions of the data collection phase of the analysis. The Expect scripts were used to interface with CIPHER<sup>®</sup> to compute the Handling Quality and Stability Margin parameters and the load motion characteristics, which was done during data processing. Example scripts used in data collection and processing are shown in Appendix C.

## IV. SIMULATION VALIDATION

Once the integration of the slung load dynamics, main rotor wake model, and the CONEX static aerodynamics model was complete, the Gen Hel/SL simulation was validated. Starting with the no-load case to ensure that the Gen Hel/SL frequency response matched flight test data, validation proceeded to a non-aerodynamic load to verify helicopter response, and finally progressed to the aerodynamic CONEX load to verify load pendulum motion.

### A. VALIDATION METHODOLOGY

As shown in Figure 18, time history data from both flight test and Gen Hel/SL simulation model were subjected to spectral analysis with CIPHER<sup>®</sup> in order to yield Bode plots for HQ, SM, and the load pendulum stability roots. From these comparisons model revisions were made as necessary. Starting with the no-load case, and using proposed FAA Level D certification criteria (Ref. [10]) for simulation frequency response fidelity as a guide, Gen Hel/SL's response in the range  $[0.5, 20]$  rad/sec was examined for both lateral and longitudinal cyclic inputs.

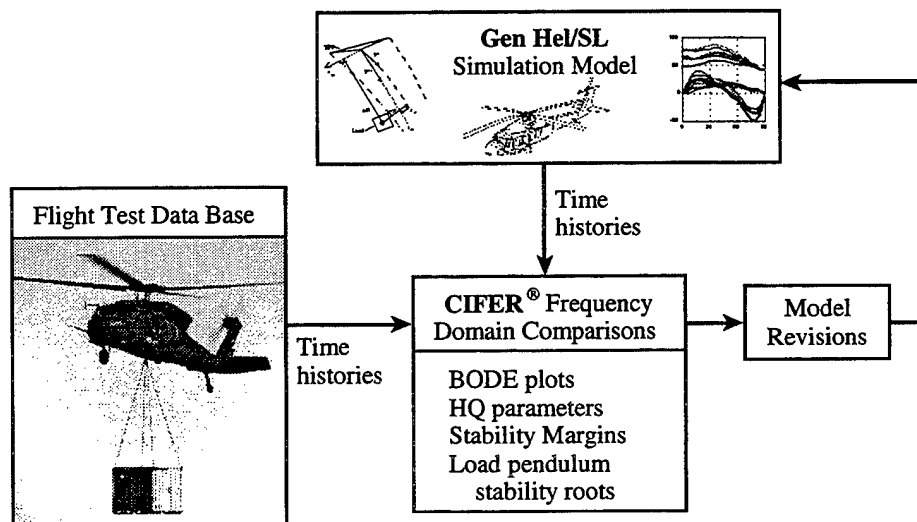


Figure 18. Validation Methodology

The fidelity requirements, or “goodness of fit,” required computing the error function by dividing the simulation frequency response by the flight test response. Identical responses yield 0 dB magnitude and  $0^\circ$  phase when divided. The resultant error function was compared with the upper and lower gain and phase boundaries to indicate frequency ranges where the simulation fell short of the realism required for piloted handling quality simulations.

The following Sections cover the no-load, 4K Block, and 4K CONEX configurations. The complete parameter results, frequency response data, and details of the system parameter determinations are consolidated in Appendices D, E, and F, respectively.

## **B. NO-LOAD SIMULATION FIDELITY**

The Gen Hel/SL and flight attitude responses and error functions for the no load condition are shown in Appendix D. The results for the hover condition showed that the lateral axis error function magnitude was within the boundaries but phase was outside the boundary above 8 rad/sec, while the longitudinal axis gain and phase were both outside the boundary at higher frequencies. Thus, Gen Hel/SL did not adequately reproduce the frequency response in the region of 2 Hz where the phase shift reaches  $-180^\circ$ , a crucial region in determining HQ parameters. Since the uncorrected Gen Hel/SL's phase shift in Figure D.1(a) fell more slowly than the flight data in this region, Gen Hel/SL yielded optimistic results.

To correct the no-load Gen Hel/SL frequency response to match as close as possible the aircraft's response, the error functions were fitted with a simple gain and time delay. This was repeated at all test airspeeds, and the results are shown in Table 5. Although correction factors were computed at each of the tested airspeeds, there was not enough data to assume that the error in the Gen Hel responses should vary with airspeed. Based upon this analysis a correction based on the average gain and time delay was added to the Gen Hel/SL frequency responses. The correction was applied during post-run processing rather than inserted into the simulation.

Airspeed (kts)	Lateral Axis		Longitudinal Axis	
	K	$\tau$ (msec)	K	$\tau$ (msec)
0	0.929	42.7	0.767	54.6
30	1.04	48.0	0.762	33.6
50	1.03	48.4	0.854	47.0
80	1.01	54.9	0.853	68.8
Average	1.00	48.5	0.809	51.0

Table 5. No Load Simulation Correction Factors

Gen Hel/SL validation was previously considered in Ref. [14] where an end-to-end time delay difference from flight data of 50 msec was computed, which was consistent with the present results. Some further comparisons with available flight data at several points in the control system were made and these indicated that the time delays were partly due to inaccuracies in the control system model and the remainder to inaccuracies in the rotor model. The primary servo actuator dynamic models have been verified so that the control portion of the delay was likely due to unmodeled linkage and mixer effects. The rotor portion of the delay was likely due to the lack of in-plane (lead-lag) structural flexing of the blades (Ref. [28]). From this point on all Gen Hel/SL HQ and SM frequency responses are given with the average correction applied.

### 1. Handling Quality

Handling Quality parameter results are collected in Figure 19 with the ADS-33D Pilot Rating level boundaries for airspeeds of {0, 30, 50, 80} kts. Values varied little over the airspeed range considered, and Gen Hel/SL essentially reproduced the flight test results, with ratings well inside the Level 1 boundary for the lateral axis, and close to the boundary for the longitudinal axis. As shown in Figure D.2, the lateral axis bandwidth was set by the frequency for 6 dB gain margin at all test speeds. Since this depends on the frequency for  $-180^\circ$  phase shift, it was affected by the correction. The longitudinal axis bandwidth was set by the  $135^\circ$  phase shift frequency, which was less sensitive to the correction function.

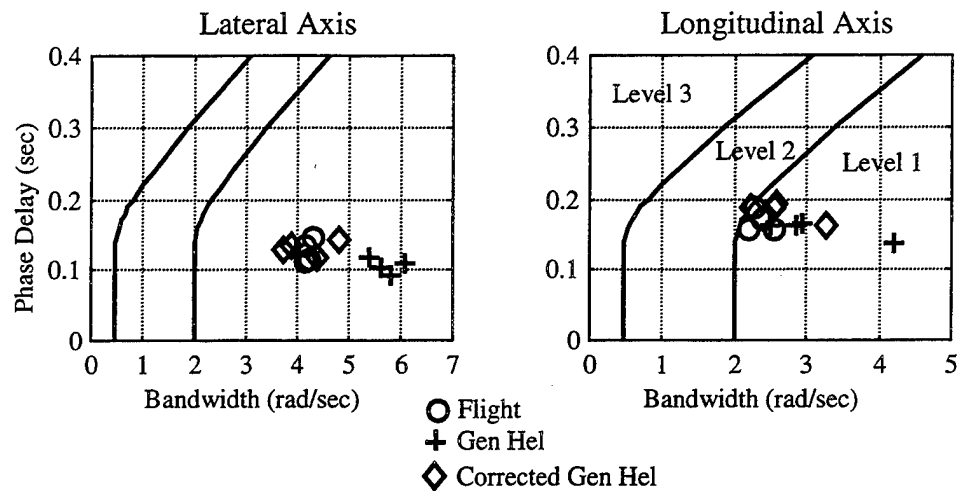


Figure 19. No Load Configuration Handling Quality Results

## 2. Stability Margins

The differences between the Direct and Indirect Methods of obtaining SM frequency responses can be seen in Figures D.6 and D.8. The differences were large at frequencies around the  $-180^\circ$  phase shift which results in large discrepancies in the computed Gain Margin. The response difference suggested unmodeled high frequency losses in the SAS linkage. The Indirect Method can be considered as the more realistic one for computing SM since it measured actual feedback to the rotor without linkage losses. In that case Gain Margins cannot be predicted accurately without improvement to the linkage simulation model, but Phase Margins can be obtained since these depend on response behavior at lower frequencies where the responses from the two methods are much closer.

Stability Margin results are shown in Figure 20 for all test speeds. Gain Margins were significantly higher from the indirect computation for both axes. Simulation Gain Margins matched flight values from the Direct Method which neglected the summing linkage dynamics and nonlinearities. These GM values were always less than the flight results from the Indirect Method so that Gen Hel yields conservative estimates. Comparison of Gen Hel to flight data in

Ref. [29] reached similar conclusions about the conservative nature of the simulation GM predictions. Phase Margins from both methods and from the simulation agreed well. The Indirect Method for computing SM is used hereafter.

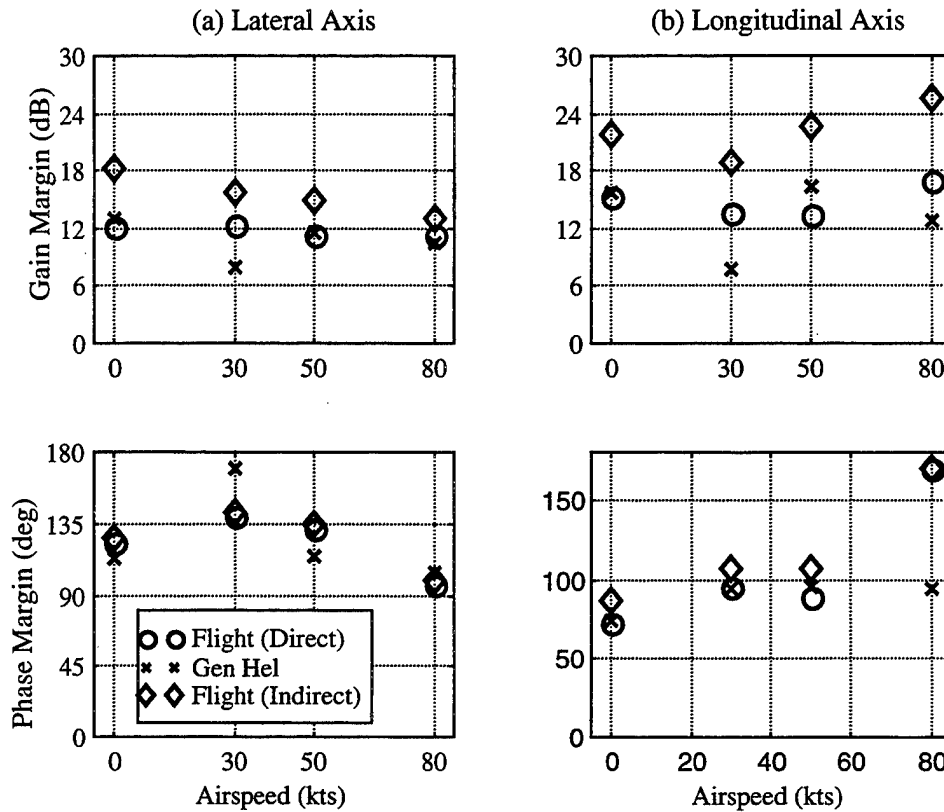


Figure 20. No Load Configuration Stability Margin Results

## C. 4K BLOCK LOAD

### 1. Handling Quality

Flight and simulation frequency responses are compared in Figures E.1 and E.2 for the lateral axis. Here the load introduced a gain dip and phase shift in the region of the pendulum mode frequency at about 1.6 rad/sec compared to the response without a load. In this region the



control inputs went into exciting the pendulum mode and less into exciting the helicopter, and there was a corresponding dip in coherence. The load also caused the magnitude plot to flatten between the pendulum frequency and about 4 rad/sec, where the response resumed the normal 20 dB/decade roll-off characteristic of rigid body dynamics. Gain and phase differences in the frequency range 6-11 rad/sec were visible and suggest some excitation of rotor dynamics by the load not captured by the simulation. The corresponding error functions were close to the limit of the Level D accuracy criteria in this range but there is nevertheless good agreement in bandwidth. Similar plots for the longitudinal axis (Figures E.3 and E.4) showed much less effect of the load on the pitch attitude response compared to the response in the lateral axis.

One result of the gain dip in the lateral axis response was that there are multiple values for the 6 dB gain margin bandwidth, one of which was just below the pendulum frequency. A similar effect was found at all test airspeeds. The question arose as to whether pilot opinion of handling qualities was correlated with either of these bandwidths. Recent unpublished Army simulation trials at NASA Ames suggested that neither of these sufficed to predict pilot opinion, and the matter of what parameter predicts pilot opinion for the slung load system remains an open question. Lateral axis results for all test airspeeds are collected in Figure 21. The simulation captured the multiple bandwidths and accurately predicted both bandwidth and phase delay in all cases, except the lower bandwidth was not captured by the simulation at 80 kts owing to small differences in response magnitude around the pendulum frequency. As with the no-load case, the lateral axis bandwidth was determined by the 6 dB gain margin frequency at all test speeds. Results for longitudinal axis handling qualities parameters are collected in Figure 22. In all cases, the  $-135^\circ$  phase shift frequency determined bandwidth.

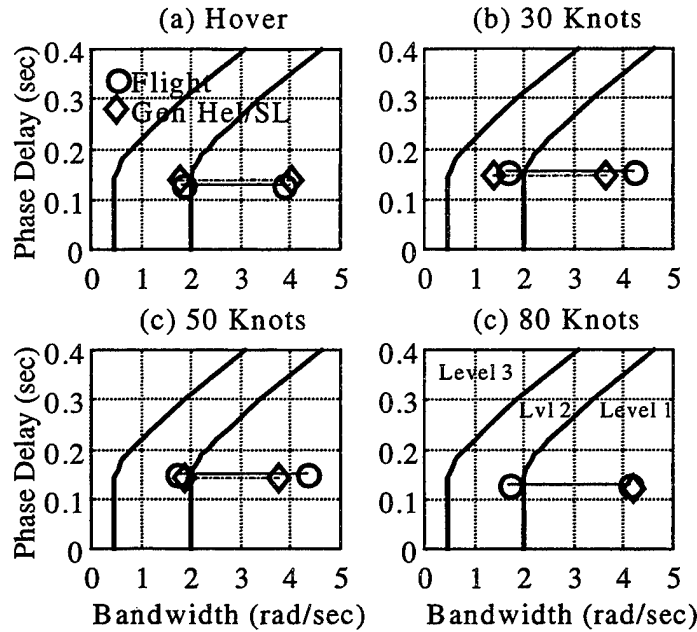


Figure 21. 4K Block Lateral Axis Handling Quality Results

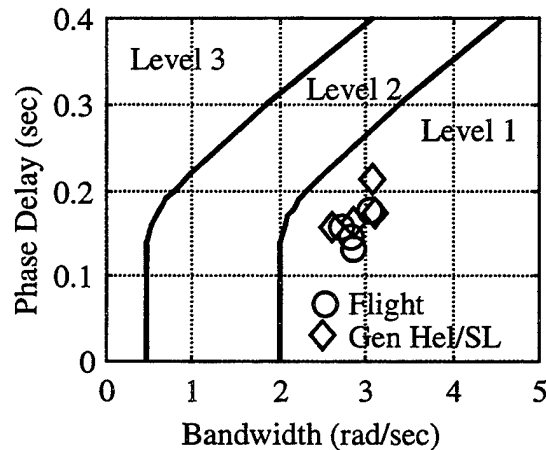


Figure 22. 4K Block Longitudinal Axis Handling Quality Results

## 2. Stability Margins

The comparison of flight and simulation results in Figure 23 for the test airspeeds, {0, 30, 50, 80} kts, which showed good agreement within the limitations of the model discussed previously. The frequency responses and details of the Stability Margin computation for both Direct and Indirect Methods are shown in Figures E.5 through E.8.

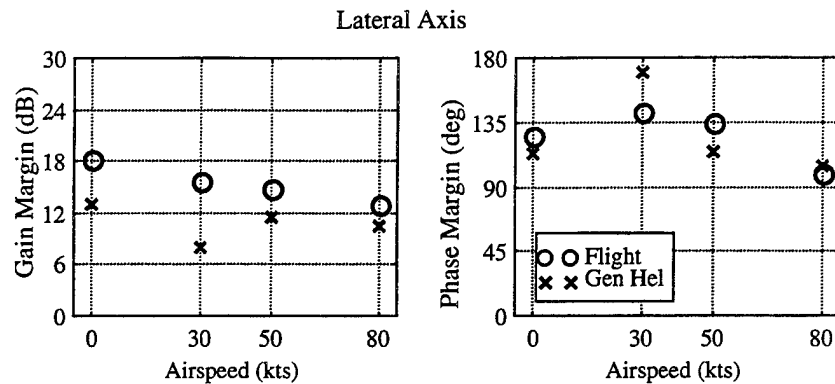


Figure 23. 4K Block Stability Margin Results

### 3. Load Pendulum Roots

The load on-axis angular rate frequency response plots for flight and simulation are compared in Figures E.9 through E.12. For the hover condition, Gen Hel/SL was seen to reproduce the flight response closely. Lateral axis coherence was good over the frequency range shown and this was the case at all test airspeeds. Longitudinal axis coherence was poorer than the lateral axis at all test speeds, including a dip below 0.6 around the pendulum frequency. The loss in coherence in the region of the gain peak suggested the presence of nonlinearities in the response.

While the nonlinearities were not yet understood, it was noted that the simulation captures the effect. The second order fit of the flight data transfer function was done over the frequency range [0.5, 2.5] rad/sec. For the lateral axis the accuracy of the fit as represented by the NAVFIT-generated cost function was high, above 100 at all test speeds, reflecting more complexity in the frequency response than can be captured by a second order transfer function model. The good coherence of the flight data tended to confirm the mismatch. For the longitudinal axis the cost of the fit was well below 100 and this reflected greater agreement between the second order pole model and the flight data. Some small order differences between

the two responses occurred, probably related to the lower coherence of the flight data, but the gross trends in gain and phase matched those of a second order pole.

Results for the longitudinal and lateral pendulum roots are collected in Figure 24 for test airspeeds to 80 kts. Load flight data was not available above 50 kts. For both modes, the natural frequency was closely predicted by Gen Hel/SL and was seen to be nearly invariant with airspeed, and virtually the same for both axes. Damping results showed good agreement at hover but differences developed with airspeed.

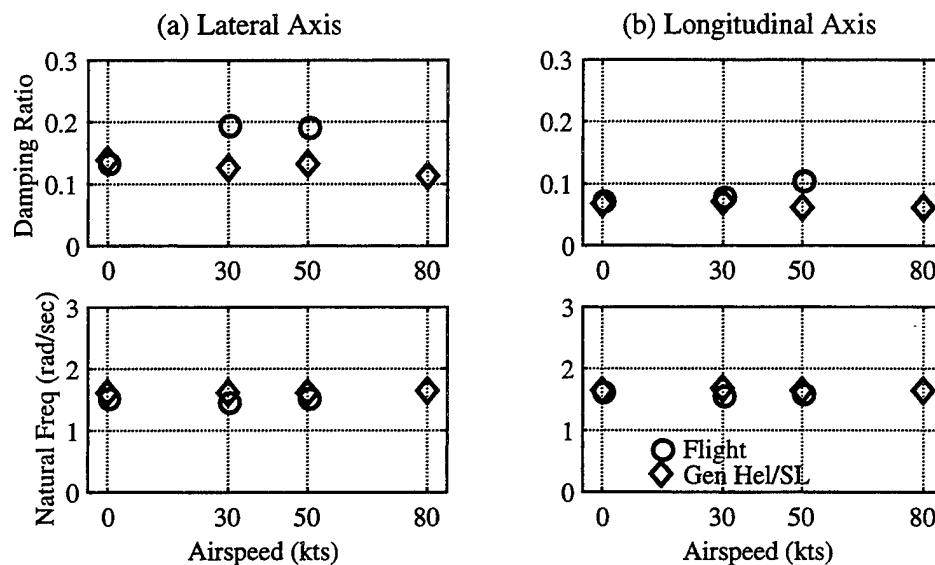


Figure 24. 4K Block Load Characteristics Results

The longitudinal pendulum was lightly damped, below 0.1, which was reflected in persistent motion in flight time histories after it was excited. The lateral pendulum was more damped, by way of its greater coupling with the aircraft attitude dynamics, and it was observed to die out in only a few cycles in flight.

## D. 4K CONEX LOAD

### 1. Handling Quality

Results for the 4K CONEX lateral axis (shown in Figure 25) are similar to those for the 4K Block load. Lateral axis bandwidth was again determined by the 6 dB Gain Margin frequency which is double valued as seen in Figures E.1 and E.2. Good agreement between flight and simulation was obtained except for the bandwidth at hover. This arose from frequency response differences in the range of 6-11 rad/sec previously noted for the 4K Block, but which resulted in a large difference in the 6 dB gain margin values and a corresponding significant bandwidth prediction error in this case. Except for the hover flight test bandwidth, parameter values were nearly independent of airspeed, nearly the same as for the 4K Block, and closely predicted by Gen Hel/SL. There was no significant effect of the CONEX static aerodynamics model on the HQ parameters.

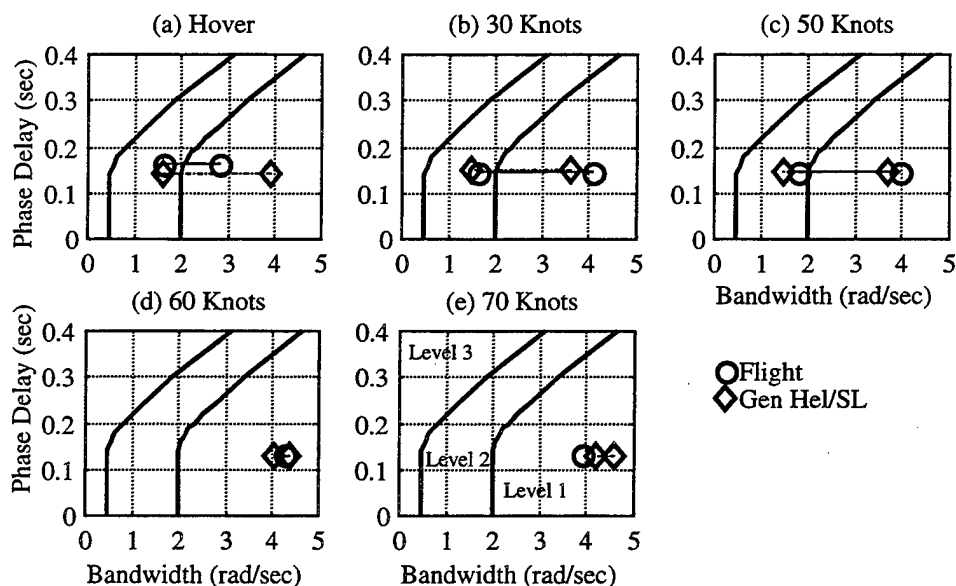


Figure 25. 4K CONEX Lateral Axis Handling Qualities Results

## 2. Stability Margins

The comparison of flight and simulation results in Figure 26 for all tests speeds {0, 30, 50, 60, 70} kts indicated good agreement for the lateral axis. For the longitudinal axis, the flight results had significantly higher Gain Margins than the simulation predicts owing to the effect of the SAS linkage model error noted earlier.

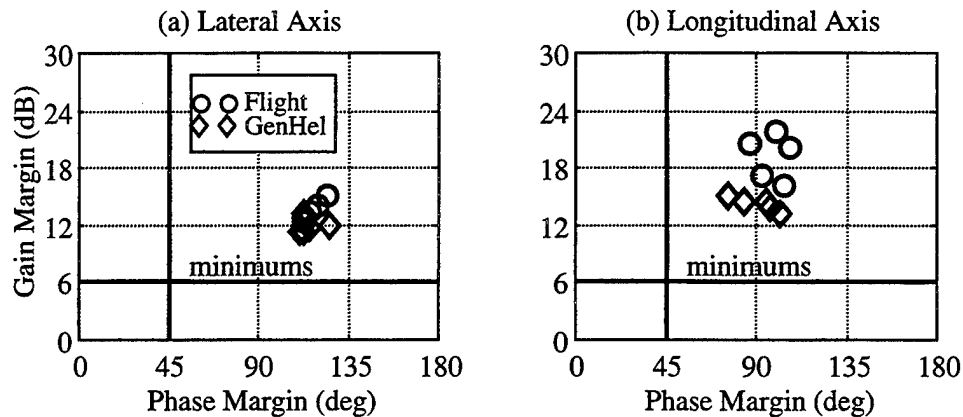


Figure 26. 4K CONEX Stability Margin Results

## 3. Load Pendulum Roots

As airspeed increases it was increasingly difficult to get load response flight data with adequate coherence for a credible identification of the pendulum roots. Coherence was insufficient above 50 kts. One difficulty was the CONEX rate of spin which increased with airspeed and which degraded the available load measurements. The second order pole fit to the flight data succeeded better than for the 4K Block, with cost below 100 in all cases.

The collected results in Figure 27 included simulation values with and without the load static aerodynamics. The pendulum frequency was seen to be accurately predicted by Gen Hel/SL. The result was insensitive to the load static aerodynamics, and nearly identical to the pendulum frequency of the block. The flight data showed a moderate increase in lateral

pendulum damping with airspeed, and the simulation predicted this if the load static aerodynamics were included. The value of damping in hover was unaffected by the rotor downwash on the load. The lateral pendulum damping of the CONEX was a little higher than for the Block, presumably due to differences in load-sling geometry details, and this increase was captured by the simulation.

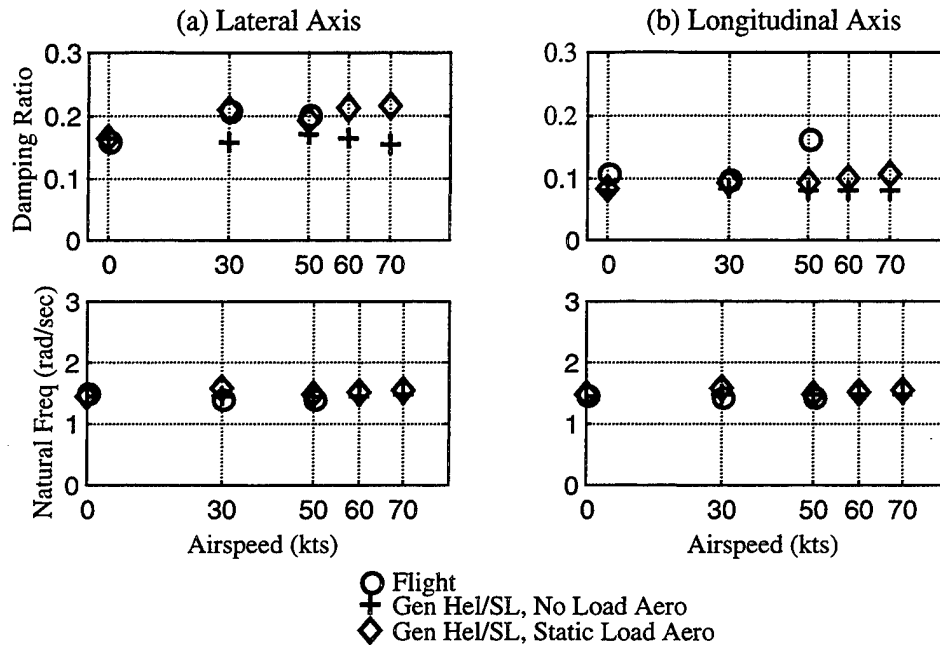


Figure 27. 4K CONEX Load Characteristics Results

## E. DATA TRENDS

The existence of trends with load weight and airspeed was considered in Figure 28 for the lateral axis HQ parameters. The simulation data include some results for a 6K lbs Block. The flight data showed little variation in either bandwidth or phase delay with airspeed or load weight, and general values of 4 rad/sec for bandwidth (using the higher of the two values for bandwidth for cases with a load) and 0.15 secs for phase delay. An exception was the moderate loss of

bandwidth at hover due to the load, with a different loss depending on the load. There was also a moderate increase in phase delay at 30 and 50 kts for the test loads. The simulation results also showed little variation with airspeed and load weight, and good general agreement with the flight values for these parameters. However, the simulation did not capture the hover loss in bandwidth due to the load and showed a sizeable difference in bandwidth at 30 kts.

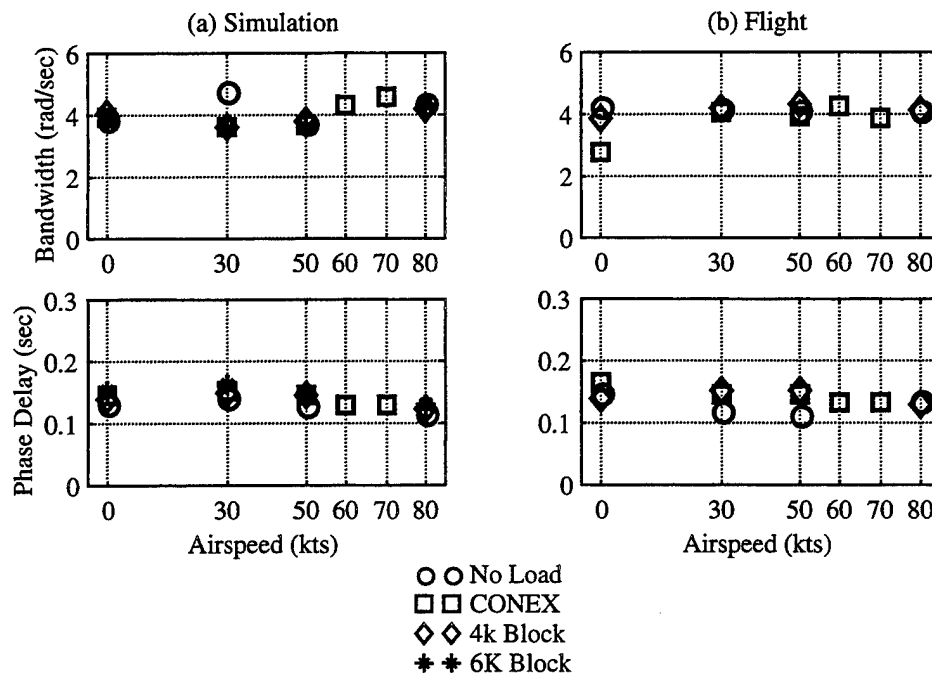


Figure 28. Effect of Load Weight on Handling Quality

Trends for lateral axis Stability Margins were considered in Figure 29. The flight results showed a differentiation in Gain Margin among loads at hover but not at higher airspeeds. The data include a 9K lbs test load result at hover from Ref. [29] which was consistent with a trend of increasing Gain Margin loss with load weight at hover. The simulation did not capture this variation and generally yielded low Gain Margins from the SAS linkage modeling error. Flight values of Phase Margin also showed consistent losses due to the load, particularly at hover. The Gen Hel/SL results for Phase Margin also showed losses for the 4K and 6K Block loads, while



there was little or no loss for the CONEX. The no-load result at 30 kts contradicted this trend but the value may be affected by marginal coherence for the simulation data at this case, which also produced an out-of-trend value for the gain margin. The CONEX computations were repeated without load aerodynamics and there was no change in the result, so the difference between results for the CONEX and the blocks was presumed due to differences in load-sling geometry between these loads. The Gain Margin results for a 6K Block were almost identical to the 4K Block.

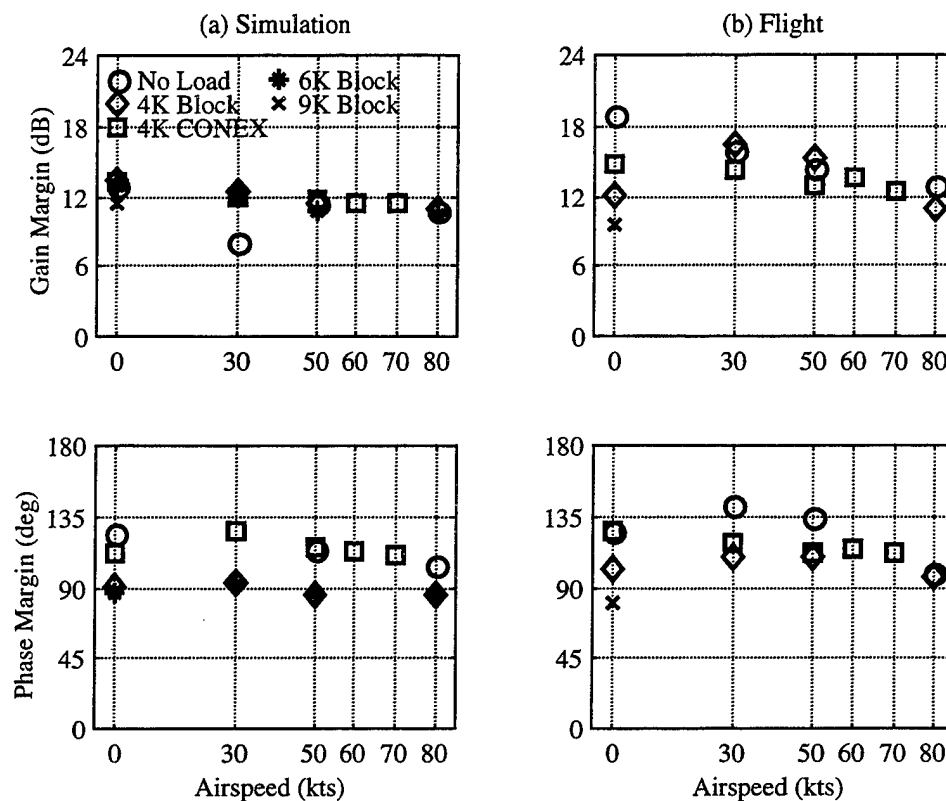


Figure 29. Effect of Load Weight on Stability Margin

## V. CONCLUSIONS

The conclusions obtained from this study are broken into two areas. First, specific observations about Gen Hel/SL's performance as a prediction tool for system dynamic parameters are discussed, followed by a general treatment of elements of this project that need further investigation.

### A. EVALUATION OF GEN HEL/SL

A slung load simulation comprised of the Gen Hel simulation model, the dynamic equations for the 2-body slung load system and multi-cable sling, and load aerodynamics due to rotor downwash and static aerodynamics, was implemented and compared with flight test data for the helicopter alone and for several test configurations at airspeeds to 80 kts.

#### 1. Simulation Discrepancies

The fidelity of the Gen Hel simulation (no load) in producing the on-axis frequency responses for the longitudinal and lateral axes over the frequency range of interest was evaluated. Additionally, the handling qualities were evaluated. Important mismatches in the attitude response used to compute Handling Quality parameters were noted in the region of 2 Hz due to control linkage and rotor dynamic modeling inaccuracies. These could be corrected empirically to obtain satisfactory agreement between simulation and flight data in attitude responses and in the Handling Qualities parameter values. Additional differences at higher frequencies were noted in the control responses used to evaluate Stability Margins which were likely due to a deficient model of the SAS linkage. These resulted in underestimated Gain Margins by the simulation. These results indicated that improvements to the Gen Hel control and rotor models would be useful in evaluating Handling Quality and Stability Margins.

## **2. Handling Quality Prediction**

Good agreement was obtained between flight data and Gen Hel/SL for the 4K lbs Block load which had negligible aerodynamic forces and moments. The simulation was able to reproduce the effects of the load on the attitude, control loop, and load angular rate frequency responses underlying the dynamic parameters of interest. The simulation reproduced such details as the multiple values of the lateral axis bandwidth, and differences between the longitudinal and lateral pendulum damping.

## **3. Stability Margin Prediction**

Gen Hel/SL showed good overall agreement with flight values for all the parameters of the test points, and captured the effects of the loads, except for Gain Margin predictions. The use of this simulation for accurate prediction of the effect of load on Stability Margins required more accurate modeling of the SAS linkage dynamics. However, the results obtained from the current model provided conservative estimates.

## **4. Load Pendulum Mode Prediction**

Good agreement was also obtained for the CONEX load, which had significant aerodynamic forces and moments. The effect of load aerodynamics on pendulum damping was obtained for airspeeds short of instability. It was found that rotor downwash had no effect on the pendulum roots, while load static aerodynamics affected damping.

## **B. AREAS FOR FUTURE INVESTIGATION**

The intent of this thesis was to evaluate a computer simulation's ability to accurately predict system parameters. In doing this an available high fidelity helicopter model was joined with previously developed slung load dynamics. Although the Gen Hel model has been

successfully used for piloted simulation and the study of helicopter performance, two areas were identified that need further work and development. With a corrected Gen Hel/SL model, the existing flight envelope can then be explored for a known load. Finally, Gen Hel/SL could explore areas of load instability and possible stabilization techniques for troublesome loads.

## **1. Model Development and Correction**

First, the lead-lag dynamics of the main rotor system model need to be improved, in order to isolate and remove a source of error. For purposes of this investigation, correction of the Gen Hel frequency responses with the addition of a time delay in the analysis was sufficient, however a more satisfactory result would be to correct the problem rather than attend to the symptoms. This has previously been done for the primary servo model in Gen Hel, which was corrected to accurately model the actual servos under flight load conditions.

Second, the SAS actuator linkage dynamics for the lateral, longitudinal, and yaw axes needs to be modeled. The current gain only model was not enough, and correction of this problem may also have beneficial effect on the rest of the simulation.

## **2. Envelope Expansion**

As stated in the Introduction, helicopter/slung load flight testing and evaluation is a costly undertaking. The Gen Hel/SL model was validated for the benchmark load of a Block without significant aerodynamic effects and the CONEX with known static aerodynamics. This was done by verifying that Gen Hel/SL could predict key artifacts such as the HQ bandwidth parameter taking on a range of values and could match the magnitudes of the load lateral and longitudinal damping ratio. With improvements as outlined above, Gen Hel/SL could be used to explore the flight envelope without requiring expensive flight testing to locate regions where the system parameters vary greatly.

This would involve multiple simulation runs, with different load and helicopter mass characteristics, longer or shorter sling lengths, different load angle of attack orientations, etc., with the goal of finding one or two configurations with significantly different system parameters. Then specific flight testing of the isolated configurations could be undertaken, to evaluate Gen Hel/SL's effectiveness in predicting the limits of the flight envelope.

An alternate method of achieving this goal could involve selecting a different aerodynamically active load, conducting wind tunnel tests to establish the static and dynamic coefficients, and then simulation to find an optimum configuration to test in flight.

### **3. Load Instability and Stabilization Methods**

An important goal of load aerodynamic models for active loads is to predict the airspeed at which the load becomes unstable due to its aerodynamics. Historically, there has not been success in modeling the aerodynamics sufficiently accurately to determine load stability, as this is likely to depend on variation of the aerodynamics with load angular rates and on unsteady aerodynamic effects.

With additional work in this area, the Gen Hel/SL simulation could be used with a known troublesome load, or with a load at high speeds or other conditions where instability is known to occur. This would involve collecting wind tunnel or flight data in a higher risk area. The goal of this testing would be to determine what methods of stabilization can be used on the load. Feedback, either to the helicopter's control system, or to a control surface mounted to the load, could be designed and incorporated into the simulation, and evaluated as to its practicality and effectiveness. In this way, the computer simulation can be used to lower risk and increase the productivity of helicopter external load operations.

## APPENDIX A: THEORETICAL BACKGROUND

### A.1. ONE AND TWO DEGREE OF FREEDOM APPROXIMATIONS

For preliminary load configuration clearance, it is useful to approximate the helicopter/slung load system with simplified models in order to find the natural frequency of load oscillation. The purpose of this approximation is to determine how close the natural frequency is to other modes in the helicopter, in this case the unstable regressive lead-lag frequency of the main rotor blades. If the two frequencies are close, then the potential exists for the main rotor system to feed energy into the load, rapidly building up oscillations to dangerous and destructive levels. As mentioned previously the main rotor system can be driven out of balance, with catastrophic results due to the air resonance.

The configurations tested in this analysis were multiple sling suspensions with relatively inelastic legs, and therefore the load is constrained such that it cannot move with respect to the sling cables. Because of this, the load motion can be estimated with a compound pendulum, with either one or two degrees of freedom (DOF) of in-plane motion. The single DOF model neglects the motion of the helicopter and treats the cargo hook as a fixed point in space and the two DOF model includes helicopter motion as a rotation about the helicopter c.g., thus including the load coupling through the hook-to-c.g. offset distance (shown in Figure A.1). The following derivation will discuss the two DOF case, and then specialize the solution to the single DOF case by setting the hook-to-c.g. offset to zero.

The kinetic and potential energies of the load are expressed as the following:

$$T = K.E. = \frac{1}{2} (m_1 v_1^2 + I_1 \dot{\theta}_1^2 + m_2 v_2^2 + I_2 \dot{\theta}_2^2) \quad (A.1)$$

$$V = P.E. = m_2 g z (1 - \cos \theta_1) + m_2 g l (1 - \cos \theta_2) \quad (A.2)$$

where:

$$v_1 = 0$$

$$v_2 = z\dot{\theta}_1 + l\dot{\theta}_2$$

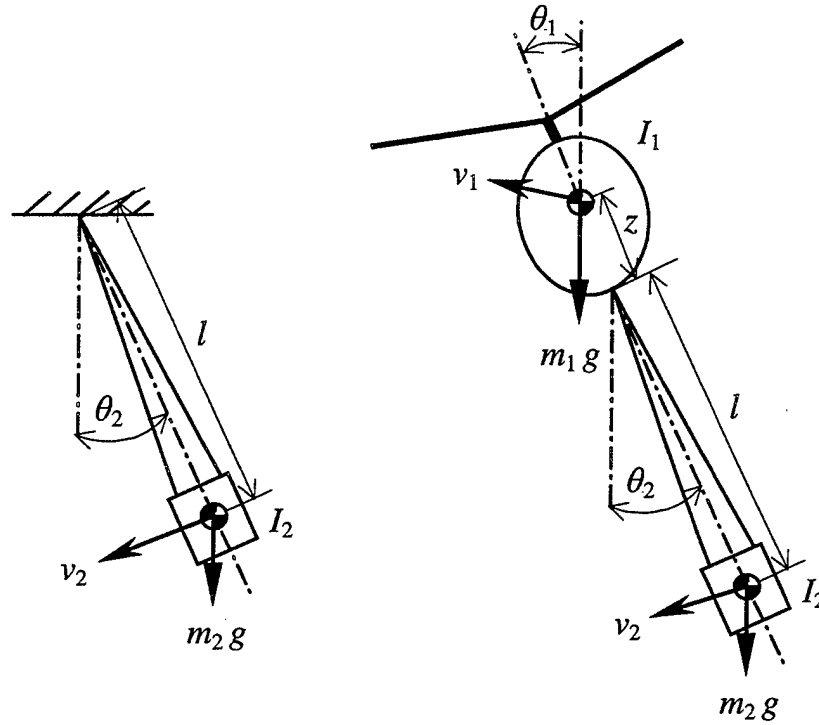


Figure A.1. One and Two DOF Pendulum Models

Using LaGrange's Equation (shown below as Equation (A.3) for a two-body system) to relate the kinetic and potential energy, and by making small angle assumptions such that  $\sin\theta = \theta$ , the equations of motion for the system are given by:

$$\frac{\partial}{\partial t} \left( \frac{\partial T}{\partial \dot{\theta}_j} \right) + \frac{\partial V}{\partial \theta_j} = 0, \quad j=1,2 \quad (\text{A.3})$$

$$\begin{aligned} [m_2 z^2 + I_1] \ddot{\theta}_1 + m_2 z l \ddot{\theta}_2 + m_2 g z \theta_1 &= 0 \\ [m_2 l^2 + I_2] \ddot{\theta}_2 + m_2 z l \ddot{\theta}_1 + m_2 g l \theta_2 &= 0 \end{aligned} \quad (\text{A.4})$$

In order to solve the equations of motion in Equation (A.4), one must find the frequency of oscillation for the system whereby each component (helicopter and load) are excited at the same frequency and reach a point of maximum excursion from equilibrium at the same instant.

In this case the solution is satisfied by the following equations:

$$\theta_1 = A_1 \cos(\omega_n t - \alpha), \quad \theta_2 = A_2 \cos(\omega_n t - \alpha) \quad (\text{A.5})$$

where  $A_1$  and  $A_2$  are the amplitudes of oscillation,  $\omega_n$  is the natural frequency, and  $\alpha$  is an unknown phase shift.

By substituting Equation (A.5) into Equation (A.4) and canceling the common factor  $\cos(\omega_n t - \alpha)$ , the result is two ordinary algebraic equations:

$$\begin{aligned} A_1 [m_2 g z - \omega_n^2 \{m_2 z^2 + I_1\}] - A_2 [m_2 z l \omega_n^2] &= 0 \\ -A_1 [m_2 z l \omega_n^2] + A_2 [m_2 g l - \omega_n^2 \{m_2 l^2 + I_2\}] &= 0 \end{aligned} \quad (\text{A.6})$$

Equation (A.6) has a trivial solution where  $A_1 = A_2 = 0$ , however the nontrivial solution is given where the determinant of the coefficients of  $A_1$  and  $A_2$  vanish. This leads to the *frequency determinant* which is given by

$$\begin{vmatrix} m_2 g z - \omega_n^2 \{m_2 z^2 + I_1\} & -m_2 z l \omega_n^2 \\ -m_2 z l \omega_n^2 & m_2 g l - \omega_n^2 \{m_2 l^2 + I_2\} \end{vmatrix} \quad (\text{A.7})$$

The solution to Equation (A.7) yields a quadratic solution in  $\omega_n^2$  which in turn gives two solutions for the *natural frequency* of the system:

$$\begin{aligned} [B_1 B_2 - m_2^2 z^2 l^2] \omega_n^4 - m_2 g (l B_1 + z B_2) \omega_n^2 + m_2^2 g^2 z l &= 0, \\ B_1 &= (m_2 z^2 + I_1), \quad B_2 = (m_2 l^2 + I_2) \end{aligned} \quad (\text{A.8})$$



$$\omega_n^2 = \frac{m_2 g}{2} \left[ \frac{(lB_1 + zB_2) \pm \sqrt{(lB_1 - zB_2)^2 + 4m_2^2 z^3 l^3}}{B_1 B_2 - m_2^2 z^2 l^2} \right] \quad (\text{A.9})$$

By setting the hook-to-c.g. offset distance  $z$  to zero, a single solution for the natural frequency arises from the two DOF solution in Equation (A.9), resulting in the one DOF approximation:

$$\omega_n = \sqrt{\frac{m_2 g l}{m_2 l^2 + I_2}} \quad (\text{A.10})$$

## A.2. SLUNG LOAD EQUATIONS OF MOTION

The generic multi-cable sling configuration with  $m \geq 3$  cables is shown in Figure A.2, along with the Newton-Euler equations for the two bodies. The notation follows that in Ref. [Error! Bookmark not defined.]. The two bodies are enumerated 1 (helicopter) and 2 (load); points on the configuration are enumerated  $a$ , (hook), 1, 2, ...,  $m$  (lift points on the load), and  $1^*$ ,  $2^*$  (c.g. locations of bodies 1, 2);  $\mathbf{V1}^*$ ,  $\mathbf{V2}^*$ ,  $\omega_1$ ,  $\omega_2$  are the inertial c.g. velocities and angular rates of the two bodies; and the applied forces are the net aerodynamic forces and c.g. moments ( $\mathbf{FA1}_1$ , ...,  $\mathbf{MA2}_2$ ), gravity ( $g$ ), and the interconnection forces and c.g. moments at the hook ( $\mathbf{FC1}_1$ , ...,  $\mathbf{MC2}_2$ ). Subscripts indicate the coordinate frame in which vectors are given ( $n$ , 1, 2 denote inertial and axes and body axes for bodies 1, 2), and transformation matrices from coordinate frame  $j$  to coordinate frame  $i$  are indicated by  $\mathbf{T}_{i,j}$  (e.g.,  $\mathbf{T}_{N1}$ ,  $\mathbf{T}_{N2}$  etc).

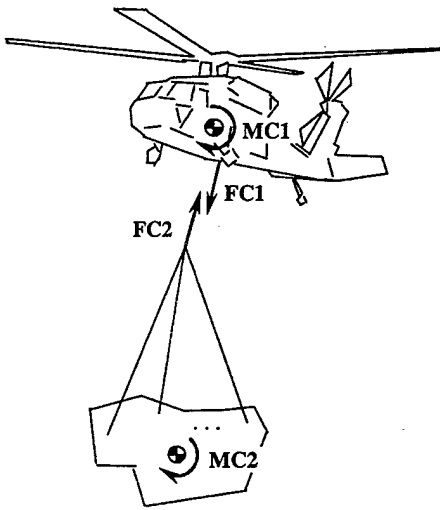
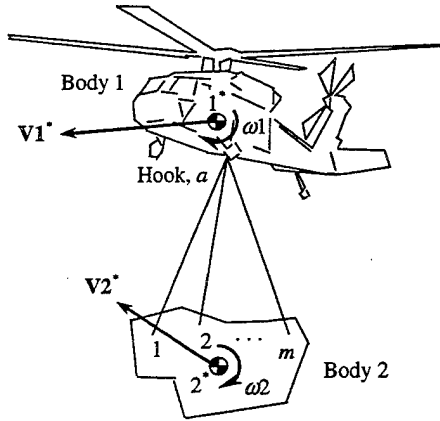
The forces and c.g. moments applied to each body at the hook are  $\mathbf{FC1}_1$ , ...,  $\mathbf{MC2}_2$ . The hook is assumed to support only force and no moments, so that these vectors can be given in terms of the three components of  $\mathbf{FC1}_N$  and other dynamic variables as seen in Figure A.2. The

notation  $\mathbf{R}_{ij}$  indicates the line segment from point  $i$  to point  $j$  in the configuration, and the skew-symmetric matrix,  $\mathbf{S}(\ )$  is used to express cross products as scalar operations.

The sling can be modeled as elastic or inelastic. Elastic cables are conventionally modeled as lightly damped springs which support only tension, in which case the hook force can be given from the cable lengths and length rates (see Figure A.2). These lengths are conveniently expressed in terms of the hook-to-load-c.g. direction vector,  $\mathbf{Ra2}^*_2$ , and the fixed geometry of the load-c.g.-to-lift-point vectors as seen in the equations. The vector  $\mathbf{Ra2}^*_2$  is an output of the system dynamics, and the parameters  $\{K_j, C_j, j = 1, \dots, m\}$  are the elastic spring and damping constants of the cables. If the cables are modeled as inelastic, then the sling applies constraints on the motion. Assume the multi-cable slings consist of three or more cables with three independent directions and then the inelastic sling imposes three constraints on the relative motion of the two bodies. In that case the three components of  $\mathbf{FC1}_N$  are dependent functions of the system dynamics as given in the figure from Ref. [12].

The equations in Figure A.2 suffice to define the two body dynamics for both elastic and inelastic multi-cable slings. The coordinates and corresponding equations are modified in the Ames implementation so that  $\mathbf{Ra2}^*_2$  and its derivatives appear as states of the system in place of the load c.g. position and its derivatives.

The parameters required for the Newton-Euler equations are the masses and inertia matrices of the two bodies,  $m1, m2, J1, J2$ . The parameters required to define the hook forces and moments are the unloaded cable lengths  $\{\ell_{0j}\}$ , the load-c.g.-to-lift-point coordinates  $\{\mathbf{R2}^*_{j2}\}$ , the elastic cable spring constants  $\{K_j, C_j\}$ , and the hook coordinates,  $\mathbf{R1}^*_{a1}$ . Parameter values for the test configurations are included in Table 2.



#### Newton-Euler Equations:

$$m1 \dot{V}1_N^* = m1 g_N + T_{N1} F A1_1 + F C1_N$$

$$m2 \dot{V}2_N^* = m2 g_N + T_{N2} F A2_2 + F C2_N$$

$$J1 \dot{\omega}1_N = M A1_1 + M C1_1 - S(\omega1_1) J1 \omega1_1$$

$$J2 \dot{\omega}2_N = M A2_2 + M C2_2 - S(\omega2_2) J2 \omega2_2$$

#### Hook Forces and Moments:

$$M C1_1 = S(R1^* a_1) T_{1N} F C1_N$$

$$F C2_N = -F C1_N$$

$$M C2_2 = S(R a2_2^*) T_{2N} F C1_N$$

#### Hook Force for Elastic Cables:

$$F C1_N = T_{N2} \sum_{j=1}^m \max \left\{ 0, K_j \left( l - \frac{\ell o_j}{\ell_j} \right) + C_j \frac{\dot{\ell}_j}{\ell_j} \right\} R a j_2$$

$$R a j_2 = R a 2_2^* + R 2^* j_2, \quad j = 1, \dots, m$$

$$\dot{\ell}_j = R a j_2^T \dot{R} a 2_2^* / \ell_j$$

#### Hook Force for Inelastic Cables:

$$F C1_N = -A^{-1} f_o$$

where:

$$A = \frac{m1 + m2}{m1 m2} I + A_{22} J1^{-1} A_{22}^{-1} + A_{23} J2^{-1} A_{23}^{-1}$$

$$f_o = \frac{1}{m1} F C1_N - \frac{1}{m2} F C2_N + A_{22} J1^{-1} M C1_1 + A_{23} J2^{-1} M C2_2$$

$$A_{22} = -T_{N1} S(R1^* a_1)$$

$$A_{23} = -T_{N2} S(R a 2_2^*)$$

$$F C1_N = m1 g_N + F A1_N$$

$$F C2_N = m2 g_N + F A2_N$$

$$M C1_N = M A1_1 - S(\omega1_1) J1 \omega1_1$$

$$M C2_N = M A2_2 - S(\omega2_2) J2 \omega2_2$$

Figure A.2. Slung Load Equations of Motion (After Ref. [12])

### A.3. COORDINATE SYSTEM TRANSFORMATIONS

In this appendix the details concerning the axes transformations used in Gen Hel/SL will be given and the resulting transformation matrices listed. All of the transformations are based on successive Eulerian rotations about the  $x$ ,  $y$ , and  $z$  axes, which describe an orthogonal coordinate system. The Euler angles are defined in accordance with standard aircraft dynamics as described in Ref. [30]. The rotations, shown in the order in which they applied through matrix multiplication, are described as:

$$\Psi = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \Theta = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}, \quad \Phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (\text{A.11})$$

Applying these rotations in proper order yields the orthogonal transformation matrix (cosine and sine have been abbreviated as  $c$  and  $s$ ):

$$\mathbf{T} = \Psi \Theta \Phi = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (\text{A.12})$$

Thus all that needs to be determined is the Euler angles describing the orientation of the different coordinate systems, which are inserted into Equation (A.12) resulting in the coordinate transform. The transformations are orthogonal, such that the inverse of each is the transpose, and so to reverse a transform the transpose is used as a new transformation matrix.

The coordinate systems used in Gen Hel/SL include the inertial frame, helicopter and load body axes, Tip Path Plane (TPP) and main rotor hub (shaft) axes, and the main rotor wake coordinate system. Transformation matrices between the different coordinate systems indicate the axes transformed to and from as subscripts to the transformation matrix,  $\mathbf{T}$ . For example, the

transform from helicopter body axes to the inertial frame of reference would be indicated by  $T_{NI}$  and its transpose, from inertial to helicopter body axes as  $T_{IN}$ .

Transformations between to axes systems for which the Euler angles are not known can be performed by first transforming to a known coordinate system and then again to the final system. As is done in the subroutine `wake`, transformations from rotor hub to wake axes can be performed by the transformation:

$$T_{WS} = T_{NW}^T T_{NI} T_{IS} \quad (A.13)$$

### 1. Helicopter Body Axes To Inertial Frame

The Euler angles from the inertial axes system,  $x_N$ ,  $y_N$ , and  $z_N$ , to the helicopter's longitudinal ( $x_1$ ), lateral ( $y_1$ ), and vertical ( $z_1$ ) axes are computed from the equations of motion contained in the `strike` subroutine every time step, and are available to compute the transformation matrices between the two systems. The matrix is stored in the common variables `A(16)` through `A(24)` each time `strike` is called, allowing other subroutines to access the current transformation as needed.

The rotation angles  $\psi_1$ ,  $\theta_1$ , and  $\phi_1$  are chosen such that the rotation  $\Psi$  aligns the heading of the aircraft with its actual azimuth,  $\Theta$  aligns the nose of the helicopter to its actual elevation, and  $\Phi$  aligns the bank of the fuselage to the actual bank angle, as shown in Figure A.3. In the chosen right-hand coordinate system for the helicopter, the  $z_1$ -axis is oriented downward, with the  $x_1$ -axis forward along the longitudinal axis of the helicopter and the  $y_1$ -axis out the right side of the aircraft along the lateral axis. From these rotation angles:

$$T_{NI} = \begin{bmatrix} c\theta_1 c\psi_1 & s\phi_1 s\theta_1 c\psi_1 - c\phi_1 s\psi_1 & c\phi_1 s\theta_1 c\psi_1 + s\phi_1 s\psi_1 \\ c\theta_1 s\psi_1 & s\phi_1 s\theta_1 s\psi_1 + c\phi_1 c\psi_1 & c\phi_1 s\theta_1 s\psi_1 - s\phi_1 c\psi_1 \\ -s\theta_1 & s\phi_1 c\theta_1 & c\phi_1 c\theta_1 \end{bmatrix} \quad (A.14)$$

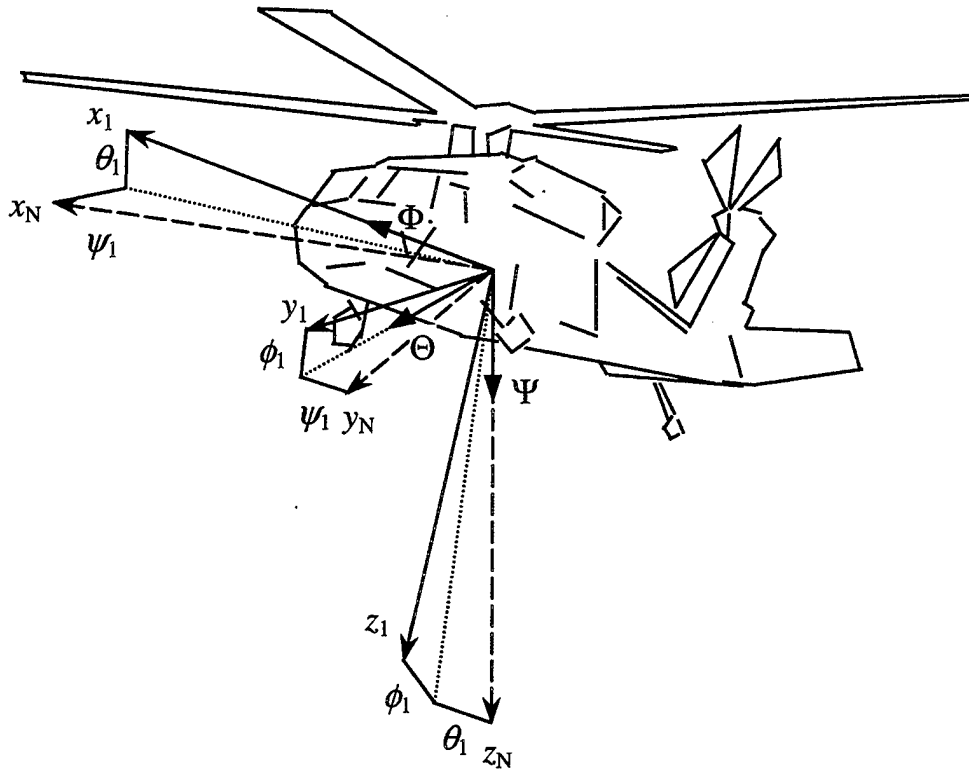


Figure A.3. Helicopter Body Axis Euler Angles

## 2. Load Body Axes To Inertial Frame

The Euler angles from the inertial axes system to the load axes,  $x_2$ ,  $y_2$ , and  $z_2$ , are computed from the equations of motion contained in the `ghslmc` subroutine every time step. The rotation angles  $\psi_2$ ,  $\theta_2$ , and  $\phi_2$  are shown in Figure A.4, along with the orientation of the inertial and load body axes. The individual rotations act in the same manner as for the helicopter as described above. From these rotation angles:

$$\mathbf{T}_{N2} = \begin{bmatrix} c\theta_2 c\psi_2 & s\phi_2 s\theta_2 c\psi_2 - c\phi_2 s\psi_2 & c\phi_2 s\theta_2 c\psi_2 + s\phi_2 s\psi_2 \\ c\theta_2 s\psi_2 & s\phi_2 s\theta_2 s\psi_2 + c\phi_2 c\psi_2 & c\phi_2 s\theta_2 s\psi_2 - s\phi_2 c\psi_2 \\ -s\theta_2 & s\phi_2 c\theta_2 & c\phi_2 c\theta_2 \end{bmatrix} \quad (\text{A.15})$$

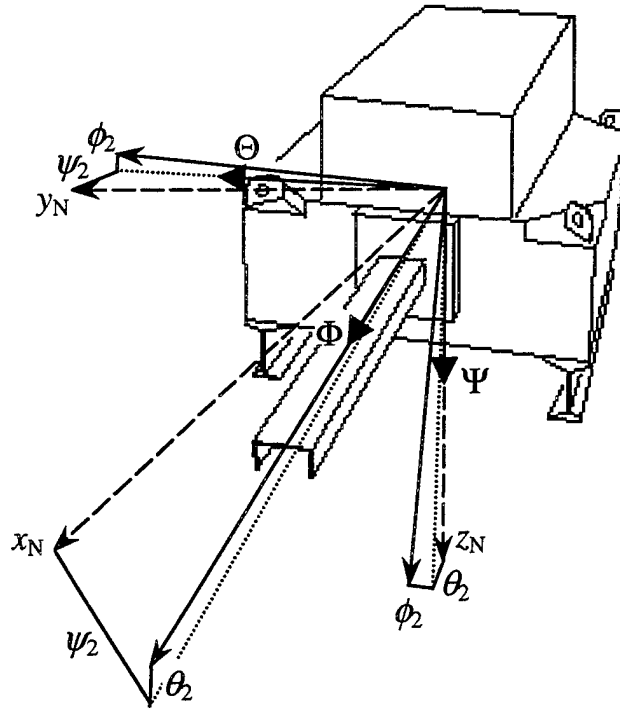


Figure A.4. Load Body Axis Euler Angles

### 3. Main Rotor Tip Path Plane Axes To Helicopter Body Axis

As part of Gen Hel's main rotor model component, the longitudinal and lateral flapping angles of the main rotor disk, relative to the main rotor hub, are computed as  $A1F$  and  $B1F$ , respectively. Adding to these angles the physical tilt of the main rotor shaft relative to the vertical axis of the helicopter, the Euler angles describing the TPP are generated. In order to keep the  $x_{TPP}$ -axis aligned with the heading of the helicopter, the rotation angle  $\psi_{TPP}$  is chosen to be zero, and the complete transformation is given as:

$$\mathbf{T}_{IT} = \begin{bmatrix} c\theta_{TPP} & s\phi_{TPP}s\theta_{TPP} & c\phi_{TPP}s\theta_{TPP} \\ 0 & c\phi_{TPP} & -s\phi_{TPP} \\ -s\theta_{TPP} & s\phi_{TPP}c\theta_{TPP} & c\phi_{TPP}c\theta_{TPP} \end{bmatrix} \quad (A.16)$$

$$\begin{aligned}
&\psi_{\text{TPP}} = 0 \\
\text{where: } &\theta_{\text{TPP}} = \text{XIS} + \text{A1F} \\
&\phi_{\text{TPP}} = \text{B1F}
\end{aligned}
\tag{A.17}$$

#### 4. Main Rotor Hub (Shaft) To Helicopter Body Axes

The transformation from main rotor hub to helicopter body axes is a special case of the TPP to helicopter body transformation where the rotor flapping angles are zero. Thus, the transformation only considers the installation angle of the main rotor transmission, given in Gen Hel/SL as XIS. In the case of the UH-60A, the main rotor shaft is tilted forward approximately 4 deg. The transformation matrix is given as:

$$\mathbf{T}_{\text{IS}} = \begin{bmatrix} c\theta_s & 0 & s\theta_s \\ 0 & 1 & 0 \\ -s\theta_s & 0 & c\theta_s \end{bmatrix}
\tag{A.18}$$

$$\text{where: } \theta_s = \text{XIS}
\tag{A.19}$$

#### 5. Main Rotor Wake To Helicopter Body Axes

The main rotor wake axes are aligned such that the vertical  $z_w$ -axis points in the direction of the flow in the center of the far-wake field. From Momentum Theory, the helicopter in a no-wind hover generates a wake which flows perpendicular to the TPP and contracts to a minimum diameter at approximately 1.5 times the rotor radius below the rotor disk. As the helicopter moves forward, the wake tends to proceed aft from the disk. This feature of the wake is also present from sideward or rearward motion, or from a wind gust, the wake proceeds along the direction of the relative wind in each case.



The induced flow at the disk, combined with the hub's translation, gives rise to the near-wake and far-wake velocities according to

$$\begin{aligned} \mathbf{V}'_N &= \mathbf{T}_{NI} \mathbf{T}_{IS} \mathbf{V}_S + \mathbf{T}_{NI} \mathbf{T}_{IT} \mathbf{V}_o \\ \mathbf{V}''_N &= \mathbf{T}_{NI} \mathbf{T}_{IS} \mathbf{V}_S + 2\mathbf{T}_{NI} \mathbf{T}_{IT} \mathbf{V}_o \end{aligned} \quad (\text{A.20})$$

where  $\mathbf{V}_S$  is the translation of the hub in shaft axes (given by  $\text{XMUXS}$ ,  $\text{XMUYS}$ , and  $\text{XMUZS}$  in Gen Hel/SL) and  $\mathbf{V}_o$  is the velocity vector for the dynamic inflow at the rotor disk, aligned with the  $z$ -axis of the Tip Path Plane (given as  $\text{XLAMDA}$  in Gen Hel/SL).

The direction of the far-wake velocity defines the wake coordinate system, as shown in Figure A.5, where the wake  $z$ -axis direction is aligned in the direction of the far-wake velocity at a distance of  $1.5R$  from the main rotor hub. From the Euler angles shown, the transformation matrix is:

$$\mathbf{T}_{NW} = \begin{bmatrix} c\theta_w & s\phi_w s\theta_w & c\phi_w s\theta_w \\ 0 & c\phi_w & -s\phi_w \\ -s\theta_w & s\phi_w c\theta_w & c\phi_w c\theta_w \end{bmatrix} \quad (\text{A.21})$$

where:

$$\begin{aligned} \theta_w &= \text{atan}\left(\frac{u''_N}{w''_N}\right) \\ \phi_w &= \text{asin}\left(\frac{-v''_N}{V}\right), \quad V = \sqrt{u''_N{}^2 + w''_N{}^2} \end{aligned} \quad (\text{A.22})$$

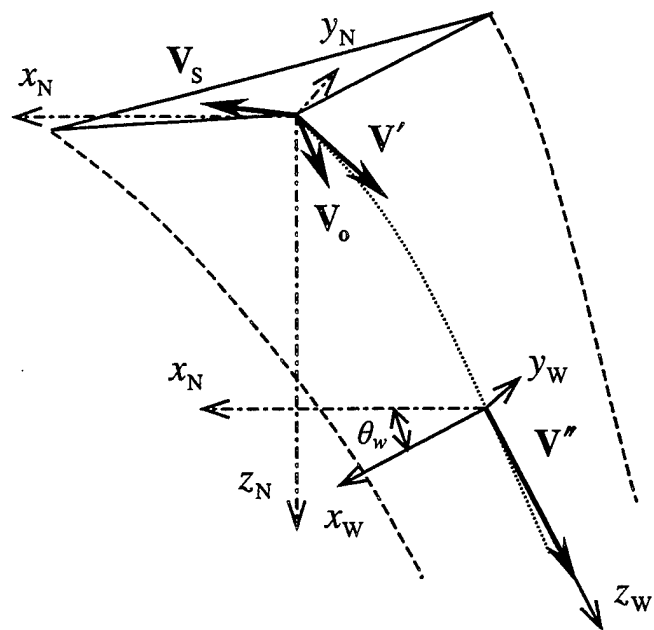


Figure A.5. Main Rotor Wake Axis Directions

## APPENDIX B. GEN HEL/SL - OPERATION AND PROGRAM LISTINGS

The Gen Hel/SL simulation primarily uses standard Gen Hel Version 6.0 subroutines, with a minimum number of changes to the original files. The slung load files are contained in the `Genhel/batch/sl` directory, along with three modified Version 6.0 files. The simulation is run from the `/batch` directory, where the `bhawk.dat` and `ghsl.dat` configuration files are located, and where the simulation run output files are saved. The command to execute Gen Hel/SL is `"/nrt/genhel"` at the command prompt.

The `makefile` located in the `/batch/nrt` directory controls the compilation of the FORTRAN executable file `genhel`, and is modified to include the additional slung load files in the compilation. The main program file, `bhawk_nrt_exec.f`, also located in the `/batch/nrt` directory, has been modified by including a call for `ghsl_init` immediately following the call for `bhawk_nrt_init`. All modifications to standard Gen Hel Version 6.0 files have the alterations marked with the comment `Cpht` at the beginning and end in the individual program listings.

Below is an index of files grouped by directory that are required for Gen Hel/SL operation. The list gives short descriptions of the slung load files and the modifications made to the three Gen Hel files that have been moved to the `/batch/sl` directory. Data automation and data processing script file names are also shown, with a full description and sample program listings of these files contained in Appendix F.

Following the directory index, detailed descriptions and program listings for the slung load files are given.

## B.1. GEN HEL/SL PROGRAM FILES

### Genhel/batch/

bhawk.dat	Standard Version 6.0 configuration file, containing parameters for a simulation run based on nominal mission profile provided by Sikorsky for ARMY 748.
fname.dat	File created by Data Automation script files to indicate the output file name of the current Gen Hel/SL case. Used by runghsl.dat.
ghsl.dat	Configuration file for the Slung Load simulation. Includes several redundant bhawk.dat variables. Called by ghsl_init.
ghsl.noload	Template file used to create ghsl.dat for no load cases. Used by /scripts/scriptcase (see description below).
ghsl.slmc	Template file used to create ghsl.dat for single lift, multi-cable suspension (slmc) cases. Used by scripts/scriptcase.
ghsl_dat.f	Data processing program used to convert FORTRAN binary output files from Gen Hel/SL to UNC3 format for CIPHER® and XPLOT analysis.
makesweep.f	Program to generate a frequency sweep input, with exponentially increasing frequency. Compiled as executable file makesweep.
sweep.dat	FORTTRAN binary data output file from makesweep, used as control history input.
sweep.xp	UNC3 format data output file from makesweep, used by XPLOT.

### Genhel/batch/nrt/

bhawk_nrt_achtrim.f	bhawk_nrt_trmswp.f
bhawk_nrt_control.f	bhawk_nrt_utils.f
bhawk_nrt_exec.f	bhawkrun.f
bhawk_nrt_init.f	engtrim.f

lastchr.f  
makefile.f  
nrt\_master\_load.f  
nrt\_store\_trends.f

nrt\_dummys.f  
nrt\_util2.f  
rndchk.f  
sweep\_on\_weight.f

Genhel/batch/rt

actdyn.f  
aero.f  
afuse.f  
amax.f  
atail.f  
bestable.f  
bdchk5.f  
bhawkblock.f  
bhprint.f  
cinput.f  
conlimit.f  
contrim.f  
dclock.f  
dinvert.f  
dummy.f  
ecut700.f  
engclu.f  
fade.f  
fafun.f  
fastp.f  
fcalls.f  
flash.f  
fps.f  
fpslgc.f  
frfun.f  
ftfun.f  
gearbox.f  
hgsmallp.f

hgtrim.f  
hmut700.f  
makefile.f  
mclosv.f  
mextend.f  
mopeno.f  
oprtn.f  
pba.f  
pdelay.f  
pfcs.f  
pointa.f  
pointv.f  
qtype.f  
rotor.f  
rtrim.f  
sas.f  
sensrs.f  
setup.f  
sfilter.f  
sofrlim.f  
stabil.f  
synchro.f  
trotor.f  
t700.f  
uh60\_indications.f  
windc.f

Genhel/batch/strike/

ardc62.f	strike.f
block.f	triang.f
btype.f	turb.f
fact.f	update.f
makefile.f	xnorm.f
still.f	

Genhel/batch/sl/

bhawk_nrt_io.f	Modified Version 6.0 file. Subroutine to controls input and output for simulation. Modifications include: <ol style="list-style-type: none"><li>1) Removal of the BHAWK_SIM_OUT//JOBSTR//.BIN output file (FLYTE system database output file)</li><li>2) Replace nrt_out with nrt_unc3_out.</li><li>3) Change the screen output from "WRITING OUTPUT DATA IN FLYTE-SYSTEM FORMAT..." to "WRITING BINARY OUTPUT FILE..."</li></ol>
conexaero.f	CONEX static aerodynamics from corrected/extrapolated Technion Wind Tunnel data.
ftotal.f	Modified Version 6.0 file. Subroutine to total the forces and moments acting on the Helicopter body from the main rotor, tail rotor, fuselage, tail, rotating parts, and downwash corrections. Modifications include: <ol style="list-style-type: none"><li>1) Addition of Slung Load variables from slvars.cmn</li><li>2) Sums the total force and moments as listed above and sends the results to either ghslsc or ghslmc.</li><li>3) Recomputes the total forces and moments including the contribution from the slung load dynamics.</li></ol>
ghsl_init.f	Variable initialization subroutine for slung load dynamics. Uses data file ghsl.dat.
ghslmc.f	Slung load dynamics for single lift, multi-cable suspension.
ghslmc_ic.f	Load-suspension initialization for slmc.

ghslsc.f	Slung load dynamics for single lift, single-cable suspension, with initialization included. Not functional.
makefile.f	Used for compilation of the executable file genhel.
nrt_master_read.f	Modified Version 6.0 file. Subroutine to read in control time history for dynamic check routine. Modifications include: <ol style="list-style-type: none"> <li>1) Add AP_RESET, a reset discrete for pilot subroutine.</li> <li>2) Disable the data searching and checking routine since sweep.dat, is of known correct format.</li> <li>3) Replace nrt_in with nrt_unc3_in.</li> <li>4) Set values for LOC_STR to indicate proper on-axis control input channel in sweep.dat.</li> <li>5) Call pilot if IPILOT = 1.</li> </ol>
nrt_unc3_in.f	Reads UNC3 formatted control history into array XMINPUT.
nrt_unc3_out.f	Writes formatted output to binary file for post-run processing.
pilot.f	Subroutine to add low-gain rate, attitude, and helicopter velocity feedback loop on roll, pitch, and yaw.
wake.f	Main rotor downwash model to compute axial and tangential flow at the load center of gravity, based on empirical data.

Genhel/batch/scripts/ scriptcase	(Data Collection) C-Shell scripts, where <i>case</i> is one of the following: <ol style="list-style-type: none"> <li>1) 4cc 4K CONEX, load static aerodynamics</li> <li>2) 4cd 4K CONEX, load drag estimation</li> <li>3) 4cn 4K CONEX, no load aerodynamics</li> <li>4) 4kd 4K Block, load drag estimation</li> <li>5) 4kn 4K Block, no load aerodynamics</li> <li>6) 6kn 6K Block, no load aerodynamics</li> <li>7) 9kn 9K Block, no load aerodynamics</li> <li>8) nl No Load</li> </ol>
-------------------------------------	---

cifer/tcltkscripts/	(Data Processing)
hq.exp	Expect script used to interface with CIPHER <sup>®</sup> for Handling Quality analysis.
hqft.case	C-Shell scripts for flight data HQ analysis, where <i>case</i> is as for <i>script.case</i> .
hqgh.case	C-Shell scripts for Gen Hel/SL simulation data HQ analysis.
hqset.tcl	Script to configure environment variables Expect script for HQ analysis, written by <i>hqft.case</i> or <i>hqgh.case</i> .
lc.exp, lcft.case,	Same as above. Load Characteristics analysis.
lcgh.case, lcset.tcl	
sm.exp, smft.case,	Same as above. Stability Margin analysis.
smgh.case, smset.tcl	



## B.2. SUBROUTINE `conexaero`

*Functions:* Develop the static forces and moments acting on the CONEX load due to the relative wind at the load center of gravity. Static Aerodynamics are from corrected/extrapolated Technion Wind Tunnel data.

*Inputs:* VA2S2 (apparent wind at load c.g., load body axes)

*Outputs:* FA22, MA22 (forces/moments due to static aerodynamics, load body axes)

*Called By:* ghslmc, ghslmc\_ic

*Calls:* None

*Comments:* Table lookup used to find drag, yaw, and yaw moment as a function of dynamic pressure. Angle of attack (-90 to +90 degrees) and yaw angle (0 to 90 degrees) are computed and used as the inputs to the tables. Yaw angles outside of range are accounted for through known symmetry and anti-symmetry of CONEX box structure. Subroutine includes provision for parametric study of the effect of dynamic load aerodynamics. This is done by finding the time rate of change of either angle of attack, yaw angle, or a combination of the two and assuming additional aerodynamic forces and moments due to the angular rate. The variable DYNAMIC is used to alter the relative effect of the dynamic forces. Set DYNAMIC to 0 in ghsl.dat to remove dynamic aerodynamic estimation.

### *Program Listing:*

```
SUBROUTINE CONEXAERO

  INCLUDE 'slvars.cmn'

  REAL ALFG(26), BETG(19), DOQT(26,19), YOQT(26,19), LOQT(26,19),
x      RMOQT(26,19), PMOQT(26,19), YMOQT(26,19), S(6,4), T2W(3,3),
x      ALF2DO, BET2DO, DALF2DO, DBET2DO

C fortran stores by column.  each column corresponds to a single value
C of beta here - the reverse of the storage in milvan_arc.f where each
C column was a value of alfa.

C range of alpha values in the tables (value for a row in the table)
  DATA ALFG/
x      -90.000, -85.000, -80.000, -75.000, -70.000, -65.000,
x      -60.000, -55.000, -50.000, -45.000, -40.000, -35.000,
x      -30.000, -25.000, -20.000, -15.000, -10.000, -5.000,
x      0.000, 5.000, 10.000, 15.000, 20.000, 25.000,
x      30.000, 90.000/

C range of beta values in the tables (value for a column in the table)
```

	DATA BETG/					
x	0.000,	5.000,	10.000,	15.000,	20.000,	25.000,
x	30.000,	35.000,	40.000,	45.000,	50.000,	55.000,
x	60.000,	65.000,	70.000,	75.000,	80.000,	85.000,
x	90.000/					

C drag force over the dynamic pressure, (row = alpha, col = beta)

	DATA DOQT/					
x	49.033,	49.801,	50.731,	54.231,	57.240,	60.197,
x	62.876,	63.852,	65.154,	67.238,	66.276,	63.165,
x	67.279,	63.825,	60.898,	58.144,	60.904,	58.944,
x	67.560,	65.488,	65.201,	65.248,	71.579,	74.108,
x	76.747,	80.306,				
x	47.031,	50.524,	50.711,	53.314,	57.111,	60.127,
x	62.879,	63.588,	64.713,	65.720,	65.126,	62.778,
x	66.235,	63.165,	59.957,	56.976,	60.498,	58.361,
x	65.590,	65.213,	65.359,	64.244,	70.734,	73.969,
x	77.355,	79.757,				
x	47.732,	50.553,	50.991,	52.637,	56.275,	59.354,
x	61.847,	61.976,	63.375,	64.536,	63.693,	61.994,
x	63.200,	60.851,	58.534,	55.829,	59.565,	58.593,
x	60.710,	64.061,	65.318,	65.009,	69.924,	73.916,
x	76.765,	78.347,				
x	49.982,	51.101,	51.900,	54.067,	56.853,	59.922,
x	61.744,	62.487,	63.956,	64.739,	64.497,	62.432,
x	63.468,	62.081,	60.276,	56.812,	60.506,	60.454,
x	62.351,	65.863,	66.554,	66.774,	70.648,	73.825,
x	75.928,	79.128,				
x	52.769,	53.505,	55.171,	56.980,	59.186,	61.252,
x	62.993,	63.464,	63.285,	65.374,	64.567,	63.308,
x	65.673,	64.743,	63.245,	60.906,	64.488,	62.395,
x	64.249,	68.454,	68.768,	69.936,	73.323,	75.460,
x	76.736,	81.487,				
x	55.360,	55.695,	58.389,	59.630,	60.954,	62.791,
x	64.146,	64.735,	64.784,	66.087,	66.665,	64.656,
x	68.327,	67.858,	66.027,	64.565,	67.582,	64.842,
x	66.576,	71.095,	72.538,	72.100,	76.194,	77.946,
x	78.760,	84.614,				
x	57.544,	57.277,	60.954,	61.874,	62.671,	64.269,
x	66.023,	66.273,	66.588,	67.619,	67.854,	65.836,
x	70.915,	70.088,	68.184,	66.754,	69.686,	67.376,
x	69.355,	73.204,	75.498,	74.600,	77.494,	78.884,
x	79.712,	86.520,				
x	58.723,	57.646,	61.174,	64.139,	64.898,	65.383,
x	66.592,	67.285,	67.947,	68.493,	68.820,	67.464,
x	72.668,	72.214,	70.158,	68.770,	70.449,	68.795,
x	70.379,	73.296,	76.725,	76.635,	78.547,	79.673,
x	81.230,	87.933,				
x	58.694,	57.780,	60.086,	64.892,	66.245,	66.139,
x	66.495,	67.356,	67.988,	69.920,	69.994,	68.833,
x	73.314,	72.532,	70.604,	69.351,	71.336,	70.164,
x	70.424,	71.062,	74.485,	76.177,	77.888,	78.847,
x	80.633,	86.804,				
x	57.919,	57.601,	58.873,	63.550,	65.275,	66.219,

x	65.571,	67.465,	67.965,	69.279,	69.187,	68.270,
x	72.282,	71.266,	69.928,	69.057,	71.317,	69.849,
x	70.052,	69.215,	70.628,	73.366,	75.958,	76.673,
x	78.052,	83.923,				
x	56.515,	56.811,	57.801,	60.951,	62.473,	63.187,
x	63.108,	64.655,	65.676,	67.231,	67.747,	66.971,
x	70.049,	69.256,	68.301,	67.426,	68.659,	67.836,
x	67.466,	66.851,	68.307,	69.435,	71.834,	72.850,
x	74.878,	79.982,				
x	53.832,	54.957,	55.586,	57.162,	57.841,	59.318,
x	60.623,	61.220,	62.905,	65.253,	65.385,	64.097,
x	66.950,	66.210,	65.753,	64.199,	66.245,	64.874,
x	64.802,	64.386,	65.620,	65.690,	67.180,	69.740,
x	71.073,	76.460,				
x	52.246,	52.668,	53.400,	54.522,	55.352,	56.725,
x	57.791,	58.632,	59.943,	61.925,	62.268,	60.549,
x	63.050,	62.603,	61.704,	60.794,	62.777,	61.643,
x	62.097,	61.513,	62.449,	62.433,	63.560,	65.167,
x	67.186,	71.743,				
x	50.691,	50.934,	51.219,	52.220,	53.239,	54.345,
x	54.676,	55.868,	56.659,	58.495,	58.517,	57.637,
x	59.348,	58.972,	58.274,	57.554,	59.022,	58.522,
x	58.810,	58.827,	60.058,	59.722,	60.524,	61.290,
x	61.938,	67.081,				
x	49.689,	49.427,	49.876,	50.639,	51.082,	51.484,
x	51.915,	52.564,	53.334,	55.152,	55.049,	54.040,
x	55.373,	55.381,	54.981,	54.447,	55.321,	55.008,
x	55.609,	56.123,	57.362,	56.634,	57.187,	57.654,
x	58.149,	62.322,				
x	48.203,	48.753,	48.812,	49.229,	49.191,	49.539,
x	49.298,	49.837,	49.731,	50.934,	50.772,	49.783,
x	51.569,	51.559,	51.467,	51.219,	51.961,	51.932,
x	52.762,	53.137,	53.950,	53.485,	53.837,	53.541,
x	53.877,	56.983,				
x	45.822,	47.332,	47.260,	46.867,	46.685,	46.997,
x	46.907,	46.350,	46.091,	46.770,	46.524,	45.912,
x	47.950,	47.937,	47.802,	48.003,	48.606,	48.626,
x	48.955,	49.273,	49.582,	49.134,	49.042,	48.635,
x	48.796,	50.801,				
x	42.410,	44.981,	44.748,	44.402,	43.879,	43.896,
x	43.485,	42.285,	42.116,	42.963,	42.938,	42.330,
x	44.205,	43.777,	43.814,	43.583,	43.976,	43.735,
x	44.060,	44.217,	44.769,	43.846,	43.701,	43.445,
x	43.717,	43.944,				
x	42.195,	42.195,	42.195,	42.195,	42.195,	42.195,
x	42.195,	42.195,	42.195,	42.195,	42.195,	42.195,
x	42.195,	42.195,	42.195,	42.195,	42.195,	42.195,
x	42.195,	42.195,	42.195,	42.195,	42.195,	42.195,
x	42.195,	42.195/				

C yaw force over the dynamic pressure, (row = alpha, col = beta)

	DATA YOQT/					
x	0.000,	0.000,	0.000,	0.000,	0.000,	0.000,
x	0.000,	0.000,	0.000,	0.000,	0.000,	0.000,

x	0.000,	0.000,	0.000,	0.000,	0.000,	0.000,
x	0.000,	0.000,	0.000,	0.000,	0.000,	0.000,
x	0.000,	0.000,				
x	10.402,	11.314,	9.775,	3.079,	0.612,	0.066,
x	-0.230,	-0.268,	0.906,	1.746,	2.087,	1.300,
x	3.879,	3.840,	4.190,	4.733,	5.545,	9.470,
x	10.937,	10.586,	11.580,	7.996,	0.636,	0.157,
x	1.398,	5.246,				
x	10.698,	14.112,	11.972,	4.804,	3.322,	2.669,
x	2.652,	2.217,	3.531,	4.714,	5.211,	5.254,
x	7.497,	8.456,	8.595,	8.846,	8.887,	14.123,
x	17.825,	17.036,	16.010,	13.333,	8.536,	5.132,
x	2.336,	12.736,				
x	10.237,	11.858,	11.140,	6.317,	5.555,	5.854,
x	6.123,	5.889,	5.982,	7.217,	8.334,	8.529,
x	10.015,	10.035,	10.414,	10.476,	13.128,	16.897,
x	18.032,	17.709,	15.685,	12.859,	11.348,	10.215,
x	7.063,	16.541,				
x	9.700,	11.228,	10.758,	9.033,	8.623,	8.449,
x	8.431,	7.757,	8.392,	8.788,	10.626,	9.814,
x	11.163,	11.582,	12.256,	12.927,	14.625,	16.295,
x	17.265,	16.315,	15.196,	13.385,	12.050,	12.459,
x	10.090,	17.323,				
x	9.667,	10.738,	11.842,	10.665,	10.427,	10.548,
x	10.637,	10.462,	10.115,	10.969,	12.009,	12.114,
x	12.607,	12.789,	13.558,	13.842,	15.136,	15.698,
x	16.548,	16.822,	16.421,	14.093,	13.593,	13.745,
x	11.921,	18.115,				
x	9.744,	11.179,	13.373,	11.909,	11.774,	11.975,
x	12.812,	12.268,	12.550,	12.619,	13.374,	13.305,
x	14.713,	14.937,	14.684,	14.267,	15.639,	15.886,
x	16.477,	17.543,	17.636,	16.179,	15.117,	14.993,
x	14.677,	19.607,				
x	11.412,	12.140,	14.601,	13.819,	13.995,	14.119,
x	14.935,	14.486,	14.386,	14.815,	15.165,	15.046,
x	16.158,	15.914,	15.652,	15.742,	16.818,	16.396,
x	16.527,	16.686,	18.118,	17.079,	16.398,	16.135,
x	17.469,	19.945,				
x	12.442,	13.680,	14.721,	15.033,	16.315,	15.863,
x	16.399,	15.551,	15.852,	16.196,	16.482,	16.086,
x	17.203,	17.644,	17.377,	16.745,	17.869,	17.220,
x	16.695,	14.854,	15.998,	17.242,	16.765,	16.063,
x	19.463,	19.265,				
x	13.042,	14.555,	14.754,	14.493,	16.426,	16.324,
x	16.138,	15.114,	15.480,	16.069,	17.412,	17.713,
x	18.477,	18.928,	18.505,	17.632,	18.522,	18.119,
x	16.253,	13.037,	13.707,	15.055,	16.155,	15.211,
x	19.879,	18.118,				
x	12.970,	14.806,	15.211,	12.831,	14.322,	15.291,
x	15.785,	14.084,	15.787,	17.381,	18.041,	18.218,
x	19.171,	19.051,	18.972,	18.298,	18.427,	17.100,
x	14.825,	12.058,	11.213,	11.241,	12.678,	12.774,
x	19.672,	15.539,				
x	11.213,	14.575,	14.931,	10.048,	11.397,	13.517,

x	15.922,	13.098,	15.355,	16.559,	16.805,	16.752,
x	18.288,	18.116,	18.465,	17.320,	17.525,	14.974,
x	13.145,	10.267,	9.122,	9.146,	9.817,	14.194,
x	19.263,	14.403,				
x	9.378,	12.986,	14.094,	9.008,	10.652,	12.450,
x	14.759,	11.069,	13.297,	14.807,	15.326,	14.020,
x	16.224,	16.255,	15.770,	14.736,	14.812,	11.958,
x	11.066,	7.859,	7.378,	7.169,	8.203,	11.180,
x	16.926,	11.129,				
x	7.923,	11.072,	11.308,	7.477,	9.362,	11.000,
x	12.108,	9.005,	11.324,	12.891,	12.744,	12.435,
x	13.545,	12.905,	12.793,	11.163,	11.977,	10.069,
x	9.516,	6.918,	6.432,	6.253,	7.625,	9.153,
x	11.287,	8.618,				
x	6.337,	9.566,	9.368,	5.589,	6.744,	7.808,
x	9.834,	6.258,	8.762,	10.595,	10.871,	9.605,
x	10.181,	10.333,	10.043,	8.475,	8.601,	7.580,
x	6.504,	4.416,	4.038,	4.280,	4.956,	5.518,
x	8.861,	5.279,				
x	5.528,	8.406,	8.595,	4.766,	5.243,	6.215,
x	7.806,	4.457,	5.963,	7.299,	8.331,	8.069,
x	9.019,	7.538,	7.613,	5.654,	6.117,	4.826,
x	4.765,	3.182,	3.217,	2.599,	2.163,	2.235,
x	6.586,	2.172,				
x	5.598,	8.156,	8.134,	4.614,	5.365,	5.948,
x	6.511,	3.340,	4.854,	6.225,	6.527,	6.013,
x	6.032,	4.378,	4.124,	2.562,	4.072,	3.847,
x	4.310,	4.099,	4.293,	3.508,	3.908,	4.395,
x	7.920,	2.962,				
x	2.796,	7.827,	7.705,	2.646,	3.450,	4.086,
x	4.347,	0.634,	1.515,	2.451,	2.565,	1.265,
x	2.726,	2.635,	2.835,	2.733,	4.479,	4.175,
x	4.899,	3.829,	4.650,	3.763,	3.307,	2.477,
x	4.432,	2.980,				
x	0.000,	0.000,	0.000,	0.000,	0.000,	0.000,
x	0.000,	0.000,	0.000,	0.000,	0.000,	0.000,
x	0.000,	0.000,	0.000,	0.000,	0.000,	0.000,
x	0.000,	0.000,	0.000,	0.000,	0.000,	0.000,
x	0.000,	0.000/				

C lifting force, rolling moment, pitching moment, incomplete

```
data LOQT/494*0./,
x   RMOQT/494*0./,
x   PMOQT/494*0./
```

C yaw moment over the dynamic pressure, (row = alpha, col = beta)

```
data YMOQT/
x   0.000,    0.000,    0.000,    0.000,    0.000,    0.000,
x   0.000,    0.000,    0.000,    0.000,    0.000,    0.000,
x   0.000,    0.000,    0.000,    0.000,    0.000,    0.000,
x   0.000,    0.000,    0.000,    0.000,    0.000,    0.000,
x   16.016,    8.590,    7.873,    10.713,    10.170,    11.732,
x   13.184,    13.505,    13.601,    15.801,    15.518,    15.163,
```

x	13.564,	13.059,	11.670,	12.105,	14.546,	16.530,
x	13.754,	12.264,	14.514,	15.574,	11.771,	7.575,
x	4.218,	11.965,				
x	28.608,	18.056,	17.488,	20.977,	20.284,	21.195,
x	20.649,	21.542,	21.284,	21.469,	19.812,	20.822,
x	21.929,	22.194,	23.319,	23.655,	26.970,	28.781,
x	32.421,	26.898,	26.081,	27.089,	26.997,	23.144,
x	11.317,	27.154,				
x	32.215,	26.988,	25.778,	30.321,	27.943,	24.917,
x	22.998,	25.484,	23.443,	25.330,	22.813,	23.372,
x	25.846,	27.589,	30.798,	31.669,	35.545,	36.404,
x	39.592,	37.140,	35.197,	34.117,	30.503,	26.826,
x	22.360,	35.409,				
x	32.101,	29.555,	30.848,	32.375,	29.519,	25.880,
x	22.953,	25.179,	22.761,	26.311,	23.632,	24.154,
x	27.303,	28.657,	31.541,	33.407,	36.099,	37.086,
x	39.346,	40.399,	37.910,	34.962,	31.552,	27.201,
x	24.182,	36.307,				
x	31.239,	29.715,	30.088,	30.697,	27.337,	23.033,
x	21.079,	22.280,	21.727,	24.965,	23.039,	23.000,
x	24.865,	26.857,	29.093,	30.232,	31.764,	33.311,
x	34.730,	37.260,	34.055,	31.275,	28.526,	25.207,
x	22.835,	31.648,				
x	27.867,	23.517,	23.194,	27.191,	23.548,	19.785,
x	17.282,	19.123,	20.298,	23.422,	21.786,	20.930,
x	23.964,	25.323,	27.205,	29.410,	27.791,	30.319,
x	29.225,	31.962,	29.731,	27.005,	24.387,	22.252,
x	20.588,	29.013,				
x	19.955,	15.679,	13.481,	23.943,	19.810,	15.740,
x	12.454,	17.280,	19.473,	23.144,	22.415,	22.116,
x	25.161,	26.355,	27.081,	27.593,	26.385,	28.855,
x	25.934,	25.658,	26.770,	25.376,	21.963,	19.785,
x	18.511,	30.233,				
x	11.938,	7.203,	7.686,	23.802,	19.595,	13.396,
x	9.469,	17.946,	20.397,	26.861,	25.698,	25.144,
x	24.533,	24.209,	24.997,	25.843,	24.493,	25.381,
x	23.200,	21.960,	22.808,	23.891,	19.557,	16.530,
x	15.800,	29.041,				
x	5.379,	0.693,	4.304,	21.571,	19.120,	14.365,
x	9.743,	21.928,	23.260,	32.169,	27.308,	23.221,
x	21.336,	21.154,	21.045,	23.932,	21.788,	21.316,
x	19.711,	18.284,	17.152,	16.997,	14.755,	11.505,
x	11.275,	22.833,				
x	-0.730,	-1.329,	0.453,	16.681,	14.242,	9.851,
x	4.896,	22.101,	22.457,	29.615,	23.902,	17.824,
x	17.211,	16.794,	17.729,	20.943,	14.492,	15.318,
x	13.201,	11.659,	8.853,	5.446,	4.618,	3.734,
x	4.806,	12.164,				
x	-7.873,	-5.993,	-7.045,	6.472,	1.005,	-1.218,
x	-4.115,	13.607,	14.176,	19.164,	18.463,	12.390,
x	10.627,	10.829,	12.845,	14.309,	6.635,	6.003,
x	3.988,	3.292,	-0.134,	-2.265,	-4.361,	-3.017,
x	-3.116,	4.837,				
x	-13.761,	-11.661,	-12.232,	-3.643,	-10.245,	-10.813,

x	-13.550,	4.407,	5.536,	12.373,	12.371,	5.215,
x	0.861,	2.606,	3.619,	4.906,	-4.809,	-5.872,
x	-6.809,	-6.909,	-9.791,	-11.362,	-13.758,	-13.354,
x	-13.409,	-7.413,				
x	-17.034,	-16.907,	-19.433,	-11.099,	-16.344,	-17.716,
x	-21.376,	-1.249,	-1.717,	6.070,	4.647,	-2.748,
x	-10.847,	-8.369,	-8.595,	-6.665,	-19.017,	-18.936,
x	-19.072,	-18.835,	-21.228,	-22.909,	-25.062,	-25.267,
x	-25.398,	-25.175,				
x	-22.402,	-24.706,	-27.019,	-18.443,	-23.005,	-25.364,
x	-29.953,	-9.988,	-10.944,	-2.787,	-5.031,	-11.232,
x	-23.354,	-21.079,	-21.084,	-19.527,	-33.867,	-33.032,
x	-33.175,	-33.444,	-34.751,	-35.652,	-36.948,	-37.376,
x	-37.411,	-43.711,				
x	-27.637,	-29.797,	-31.506,	-24.463,	-29.628,	-31.474,
x	-36.701,	-19.558,	-20.717,	-10.718,	-13.090,	-19.511,
x	-32.741,	-32.247,	-32.932,	-31.339,	-47.116,	-46.560,
x	-46.802,	-47.745,	-48.451,	-48.549,	-49.417,	-49.095,
x	-47.874,	-62.491,				
x	-28.716,	-30.373,	-32.434,	-27.737,	-32.074,	-33.053,
x	-37.803,	-25.426,	-26.385,	-17.091,	-19.413,	-24.260,
x	-36.511,	-38.425,	-38.304,	-37.213,	-53.440,	-53.624,
x	-53.162,	-53.977,	-55.111,	-55.048,	-54.288,	-52.134,
x	-48.326,	-70.391,				
x	-23.360,	-24.366,	-25.909,	-22.389,	-23.951,	-24.812,
x	-27.232,	-22.881,	-22.768,	-16.573,	-17.642,	-20.793,
x	-27.955,	-29.695,	-29.409,	-29.058,	-38.110,	-38.001,
x	-37.438,	-37.017,	-37.775,	-37.516,	-36.561,	-34.945,
x	-32.383,	-45.937,				
x	0.000,	0.000,	0.000,	0.000,	0.000,	0.000,
x	0.000,	0.000,	0.000,	0.000,	0.000,	0.000,
x	0.000,	0.000,	0.000,	0.000,	0.000,	0.000,
x	0.000,	0.000,	0.000,	0.000,	0.000,	0.000,
x	0.000,	0.000/				

C (alfao,betao) is the point in 1st quadrant, [-90,90]x[0,90] with the  
C same aero as (alf2d,bet2d) except for signs. The matrix S(6,4)  
C corrects signs. S was originally defined for angles (alfa,beta) in  
C [-90,90]x[-180,180] and imposes symmetry/antisymmetry about  
C beta = {-90,0,90}.

C The current logic takes (alfa, beta) in [-180, 180]x[-90, 90]. It  
C imposes symmetry of D, Y, YM (FA2W(1), FA2W(2), MA2W(3)) about alfa =  
C (-90, 90). Only the 4th col of S is used (QUAD is only 1 or 4) and  
C thus imposes sym/antisym about beta = 0. The symmetry in alfa is  
C taken from the MILVAN and hasn't been proven true for the CONEX as a  
C general rectangular box nor as a box with appendages, and also  
C doesn't apply to components L, RM, YM (FA2W(3), MA2W(1), MA2W(2)).  
C This needs more work, but results are reasonable. 10 sep 98

C The matrix S is stored by column and is 6 rows by 4 cols

```
DATA S/
x 1., 1., 1., 1., 1., 1.,
x 1.,-1.,-1 , 1.,-1.,-1.,
```

```

x      1., 1.,-1.,-1.,-1., 1.,
x      1.,-1., 1.,-1., 1.,-1./

VA22  = VA2S2(1)**2 + VA2S2(2)**2 + VA2S2(3)**2
Q2    = 0.5*RHO*VA22

C (alf2d, bet2d) in range (-180,180] x [-90,90]
ALF2  = atan2(VA2S2(3),VA2S2(1))
BET2  = 0.
if (VA22 .gt. 0.0001) BET2 = asin(VA2S2(2)/sqrt(VA22))
ALF2D = ALF2*R2D
BET2D = BET2*R2D

C (alfao, betao) = equivalent (alfa, beta) in [-90,90] x [0,90]
QUAD  = 1
ALFAO = ALF2D
BETAO = abs(BET2D)
if (abs(ALF2D) .gt. 90.) ALFAO = sign(1.,ALF2D)*180. - ALF2D
if (BET2D .lt. 0) QUAD = 4

C aero components at (alfao, betao) in quadrant 1 = [-90,90] x [0,90]
C table lookup routines for VMS not used here. Multiply by component
C of S to correct signs as described above.
CALL serchl(ALFAO, ALFG, 26, ix, sigx)
CALL serchl(BETAO, BETG, 19, iy, sigy)
FA2W(1) = -Q2*f2d(ix, iy, sigx, sigy, 26, DOQT) *S(1,QUAD)
FA2W(2) =  Q2*f2d(ix, iy, sigx, sigy, 26, YOQT) *S(2,QUAD)
FA2W(3) = -Q2*f2d(ix, iy, sigx, sigy, 26, LOQT) *S(3,QUAD)
MA2W(1) =  Q2*f2d(ix, iy, sigx, sigy, 26, RMOQT)*S(4,QUAD)
MA2W(2) =  Q2*f2d(ix, iy, sigx, sigy, 26, PMOQT)*S(5,QUAD)
MA2W(3) =  Q2*f2d(ix, iy, sigx, sigy, 26, YMOQT)*S(6,QUAD)

C Dynamic force computation - assess range of alfdot, betdot
C Set alfdot, betdot to previous value if difference between new value
C and previous value is greater than 10 deg/sec
DALF2D = abs(ALF2D - ALF2DO)/DT
IF (abs(DALF2D - DALF2DO).GT.10) THEN
    DALF2D = DALF2DO
ELSE
    DALF2DO = DALF2D
ENDIF
DBET2D = abs(BET2D - BET2DO)/DT
IF (abs(DBET2D - DBET2DO).GT.10) THEN
    DBET2D = DBET2DO
ELSE
    DBET2DO = DBET2D
ENDIF
DANG2D = sqrt(DALF2D**2 + DBET2D**2)

C make drag force over dynamic pressure proportional to combined change
C in angle
FDALF2D = Q2*DALF2D*DYNAMIC
FDBET2D = Q2*DBET2D*DYNAMIC
FDANG2D = Q2*DANG2D*DYNAMIC

```



```

        ALF2DO = ALF2D
        BET2DO = BET2D

C add in dynamic components only during operate mode
        IF (IMODE .EQ. 1) THEN
            FA2W(1) = FA2W(1) - FDANG2D
        ENDIF
C end dynamic force computations

C compute wind axes to body coords Euler angles and transform
        CALF      = cos(ALF2)
        SALF      = sin(ALF2)
        CPSW      = cos(-BET2)
        SPSW      = sin(-BET2)
        T2W(1,1) = CALF*CPSW
        T2W(2,1) = -SPSW
        T2W(3,1) = SALF*CPSW
        T2W(1,2) = CALF*SPSW
        T2W(2,2) = CPSW
        T2W(3,2) = SALF*SPSW
        T2W(1,3) = -SALF
        T2W(2,3) = 0.
        T2W(3,3) = CALF

        FA22(1) = T2W(1,1)*FA2W(1)+T2W(1,2)*FA2W(2)+T2W(1,3)*FA2W(3)
        FA22(2) = T2W(2,1)*FA2W(1)+T2W(2,2)*FA2W(2)+T2W(2,3)*FA2W(3)
        FA22(3) = T2W(3,1)*FA2W(1)+T2W(3,2)*FA2W(2)+T2W(3,3)*FA2W(3)
        MA22(1) = T2W(1,1)*MA2W(1)+T2W(1,2)*MA2W(2)+T2W(1,3)*MA2W(3)
        MA22(2) = T2W(2,1)*MA2W(1)+T2W(2,2)*MA2W(2)+T2W(2,3)*MA2W(3)
        MA22(3) = T2W(3,1)*MA2W(1)+T2W(3,2)*MA2W(2)+T2W(3,3)*MA2W(3)
        RETURN
        END

C locate independent variable x in array tx, for table lookup

        SUBROUTINE SERCH1(x, tx, nx, ix, sigx)
        DIMENSION tx(1)
        x1 = AMAX1(tx(1),AMIN1(x,tx(nx)))
        i = 1
1       IF (i .EQ. nx) THEN
            ix = nx - 1
            sigx = 1
        ELSE IF (x1 .LT. tx(i+1)) THEN
            ix = i
            sigx = (x1 - tx(i))/(tx(i+1)-tx(i))
        ELSE
            i = i + 1
            GO TO 1
        END IF
        RETURN
        END

C 2-D table lookup routine.

```

```

FUNCTION F2D(ix, iy, sigx, sigy, nx, tf)
DIMENSION tf(1)
k01 = nx*iy + ix
k11 = k01 + 1
k00 = k01 - nx
k10 = k00 + 1
f00 = tf(k00)
f01 = tf(k01)
f10 = tf(k10)
f11 = tf(k11)
sigys = 1 - sigx
IF (sigy .LE. sigys) THEN
    F2D = f00 + (f10 - f00)*sigx + (f01 - f00)*sigy
ELSE
    F2D = f11 - (f11 - f01)*(1 - sigx) - (f11 - f10)*(1 - sigy)
END IF
RETURN
END

```

### B.3. DATAFILE **ghs1.dat**

*Functions:* Provide data input for use with compiled non-real-time Gen Hel/SL program.  
Allow user inputs for helicopter takeoff weight and fuel load, simulation run options, helicopter/load configuration parameters, and load specifications.

*Contents:* ghs1.dat namelists including the following variables:

#### HCDATA:

TOW	helicopter take-off weight (lbs)
XMOMTO	x-axis moment arm based on TOW with full fuel
FWT	current fuel weight (lbs)
FWMX	maximum fuel weight at takeoff
AIRSPEED	Trim airspeed in knots

#### SLRUN:

CHFILE	control history input file name
AXIS	(1) lateral, (2) longitudinal, (3) collective, (4) directional (only required for sweep.dat input control history file)
STRETCH	(0) inelastic cables, (1) elastic cables
NSTORE	integer value for data decimation (normally 1)
IAERSL	(0) no load aerodynamics, (1) load drag estimation only, (2) CONEX static aerodynamic model
ILOAD	(0) no-load, (2) SLMC
IPILOT	(0) no feedback for control inputs, (1) feedback included
IWAKE	(0) no wake model, (1) with wake model
ISWIRL	(0) no swirl in wake model, (1) swirl in wake
IDATA	(0) minimum data required for HQ/SM/Load Motion analysis, (1) full data output
DYNAMIC	Dynamic load aerodynamic factor, can be varied for parametric study, or set to 0 for no dynamic effect.

SLDATA: LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ, KS, CS, LCO,  
RA2PO2, R2P2SO2, R2P2S2, NC, R2PJ2, R2S12, DOQ,

# DELPS20

*Read By:* ghsl\_init

*Comments:* This file is used by Gen Hel/SL in a similar fashion to the bhawk.dat file. The parameters in the namelists are read in to the program one time only, during program initialization. The data in ghsl.dat supercedes several of the parameters contained in bhawk.dat since it is read afterwards. This allows the user to vary key parameters between simulation runs by only editing one data file.

## Program Listing:

C ghsl.dat ----- 31 AUG 98 Peter Tyson

C Data input for GenHel/SL simulation, read by GHSL\_INIT.

C To change case, comment &SLDATA line for all except desired.

&HCDATA

TOW = 14601.0,  
XMOMTO = 5307900.0,  
FWT = 1980.0,  
FWMX = 2360.0,  
AIRSPEED = 30.0,

&END

&SLRUN

CHFILE = 'sweep.dat',  
AXIS = 1,  
DATAFILE = 'GHA4C03.3',  
DYNAMIC = 0,  
STRETCH = 0, NSTORE = 1,  
IAERSL = 2, ILOAD = 0,  
IPILOT = 1, IWAKE = 1,  
ISWIRL = 1, IDATA = 1,

&END

&SLDATA

LOADNAME = '4K CONEX (BALLASTED) WITH INST PKG',  
W2 = 4105.0,  
I2XX = 1876.0, I2YY = 1482.2,  
I2ZZ = 1376.0, I2XZ = 0.0,  
KS = 9645.0,  
CS = 22.0,  
RA2PO2 = 0.0, 0.0, 18.3036,  
R2P2SO2 = 0.0, 0.0, 1.38,  
R2P2S2 = 0.0, 0.0, 1.38,  
NC = 4,  
R2PJ2 = 2.8073, -4.0626, -3.2032,  
2.8073, 4.0626, -3.2032,  
-2.8073, -4.0626, -3.2032,  
-2.8073, 4.0626, -3.2032, 12\*0,  
DOQ = 50.0,

&END

#### B.4. PROGRAM `ghsl_dat`

**Command:** `runghsl.dat` at command prompt

**Functions:** This program is used after completion of a Gen Hel/SL simulation run in order to convert the FORTRAN binary data output files into UNC3 (GETDATA) format compatible with NASA Ames programs CIPHER<sup>®</sup> and XPLOT. Compiled as executable `runghsl.dat`.

**Inputs:** Gen Hel/SL data output `filename.dat`, where *filename* is given by DATAFILE

**Outputs:**

<code>filename.out</code>	ASCII text file containing simulation options and settings, load configuration information, and trim values.
<code>filename.xp</code>	UNC3 binary file containing simulation data for each time step, customized to the load configuration (no load, single sling, or multiple legs).
<code>filename.wake.xp</code>	UNC3 binary file containing main rotor wake simulation data for each time step (only when <code>IWAKE = 1</code> )

**Calls:** `openW`, `fWrite`, `closeW` (GETDATA functions)

**Comments:** Post run processing was chosen to allow the FORTRAN binary output data to be converted into chosen format depending on analysis programs to be used. Variables saved in radian measure are converted to degrees. Program automatically detects if the simulation run was no load, `slsc`, or `slmc` and processes the appropriate variables for each case. If `IDATA` is false during a `slmc` case run, the minimum number of variables required for data processing is used to reduce the size of output data files. During automatic data collection, `runghsl.dat` looks in `fname.dat` for the current Gen Hel/SL output file name. Program can be modified to prompt for user input filename.

##### *Program Listing:*

```
C ghsl_dat.f ----- 31 AUG 98, Peter Tyson

C      Data print and plot storage file for GenHel/Slung Load
C      simulation.  Parameters nchanx = number of variables stored
C      in dat array for (1) no-load, (2) slsc, and (3) slmc cases.

      PROGRAM GHSL_DAT

      PARAMETER(NCHAN1 = 43)
```

```

PARAMETER(NCHAN2 = 126)
PARAMETER(NCHAN3 = 111)
PARAMETER(NCHAN4 = 17)
PARAMETER(NCHANW = 53)
PARAMETER(NCHANX = 21)

REAL FA11(3), FA1N(3), MA11(3), FC11(3), FC1N(3), MC11(3),
X   FA22(3), FA2N(3), MA22(3), FC22(3), FC2N(3), MC22(3),
X   R1SN(3), V1SN(3), R1SA1(3), VA2SN(3), R2SJ2(24), RA2S2(3),
X   RA2PO2(3), R2P2SO2(3), R2P2S2(3), R2PJ2(3,8), R2S12(3),
X   V1S1(3), QO(12), Q(12), U(12), LCO, LC, KS, I1XX, I1YY,
X   I1ZZ, I1XZ, I2XX, I2YY, I2ZZ, I2XZ, C(4), TAUJ(8), LCJ(8),
X   LCJO(8), DHOOK, HHOOK, DAT1(NCHAN1), DAT2(NCHAN2),
X   DAT3(NCHAN3), DAT4(NCHAN4), DATW(NCHANW), DATX(NCHANX),
X   DALF2D, DBET2D, DANG2D, ALF2D, BET2D, R1S2SN(3), DYNAMIC

REAL*8 TIME, DAT81(NCHAN1), DAT82(NCHAN2), DAT83(NCHAN3),
X   DAT84(NCHAN4), DAT8W(NCHANW), DAT8X(NCHANX)

INTEGER STRETCH, JRTD1(12), JRTD2(31), JRTD3(26), JRTD4(5),
X   NREC, AXIS
INTEGER*4 UNIT
LOGICAL*4 openW, L

CHARACTER NAME*30, FIN*30, FOUT*30, FXP*30, SN1(NCHAN1)*16,
X   SN2(NCHAN2)*16, SN3(NCHAN3)*16, SN4(NCHAN4)*16,
X   SNW(NCHANW)*16, SNX(NCHANX)*16, LOADNAME*40, CABLES
X   CHFILE*40

DATA UNIT/3/,
X RTD, G/57.2957795, 32.174/,
X JRTD1/5,6,7,11,12,13,14,15,16,17,18,19/,
X JRTD2/5,6,7,8,9,10,18,19,20,21,22,23,27,28,29,30,31,32,
X   33,34,38,39,40,41,42,43,44,45,95,96,97/,
X JRTD3/5,6,7,8,9,10,17,18,19,20,21,22,26,27,28,29,30,31,
X   35,36,37,38,39,40,44,45/,
X JRTD4/2,3,4,12,13/

DATA SN1/
X 'T1'      , 'dv1snx' , 'dv1sny' , 'dv1snz' , 'dp1'      ,
X 'dq1'      , 'dr1'    , 'v1snx' , 'v1sny' , 'v1snz'    ,
X 'p1'       , 'q1'     , 'r1'    , 'ph1'   , 'th1'      ,
X 'ps1'      , 'dph1'   , 'dth1'  , 'dps1'  , 'r1snx'    ,
X 'r1sny'    , 'r1snz'   , 'fallx' , 'fally' , 'fallz'    ,
X 'LA1'      , 'MA1'    , 'NA1'   , 'da'    , 'db'       ,
X 'dc'       , 'dp'     , 'RSAS'  , 'PSAS'  , 'YSAS'     ,
X 'DMIXA'    , 'DMIXB'   , 'DMIXC' , 'DMIXP' , 'PSFWD'    ,
X 'PSAFT'    , 'PSLAT'   , 'PSTR'  , /

DATA SN2/
X 'T'        , 'dv1snx' , 'dv1sny' , 'dv1snz' , 'dp1'      ,
X 'dq1'      , 'dr1'    , 'dp2'    , 'dq2'    , 'dr2'      ,
X 'dvalcx'   , 'dvalcy' , 'ddlc'   , 'tau/w1' , 'v1snx'    ,
X 'v1sny'    , 'v1snz'   , 'p1'     , 'q1'     , 'r1'       ,

```

```

X 'p2'      , 'q2'      , 'r2'      , 'valcx'   , 'valcy'   ,
X 'dlc'     , 'dph1'    , 'dth1'    , 'dps1'    , 'dph2'    ,
X 'dth2'    , 'dps2'    , 'dphc'    , 'dthc'    , 'rlsnx'    ,
X 'rlsny'   , 'rlsnz'   , 'ph1'     , 'th1'     , 'ps1'     ,
X 'ph2'     , 'th2'     , 'ps2'     , 'phc'     , 'thc'     ,
X 'lc'      , 'r2snx'   , 'r2sny'   , 'r2snz'   , 'v2snx'   ,
X 'v2snx'   , 'v2snz'   , 'dv2snx'  , 'dv2sny'  , 'dv2snz'  ,
X 'rsnx'    , 'rsny'    , 'rsnz'    , 'vsnx'    , 'vsny'    ,
X 'vsnz'    , 'delrsx'  , 'delrsy'  , 'delrsz'  , 'delvsx'  ,
X 'delvsy'  , 'delvsz'  , 'dldvsx'  , 'dldvsy'  , 'dldvsz'  ,
X 'ssfnx'   , 'ssfnz'   , 'ssfnz'   , 'issfnx'  , 'issfnz'  ,
X 'issfnz'  , 'iissfx'  , 'iissfy'  , 'iissfz'  , 'fallx'   ,
X 'fally'   , 'fallz'   , 'falnx'   , 'falny'   , 'fainz'   ,
X 'fa2nx'   , 'fa2ny'   , 'fa2nz'   , 'ma1lx'   , 'mally'   ,
X 'ma1lz'   , 'ma22x'   , 'ma22y'   , 'ma22z'   , 'pc'      ,
X 'qc'      , 'rc'      , 'lcmlco'  , 'rhk2s1x' , 'rhk2s1y' ,
X 'rhk2s1z' , 'fc1lx'   , 'fc1ly'   , 'fc1lz'   , 'LC1'     ,
X 'MC1'     , 'NC1'     , 'da'      , 'db'      , 'dc'      ,
X 'dp'      , 'phch'    , 'thch'    , 'rphc'    , 'rthc'    ,
X 'RSAS'    , 'PSAS'    , 'YSAS'    , 'DMIXA'   , 'DMIXB'   ,
X 'DMIXC'   , 'DMIXP'   , 'PSFWD'   , 'PSAFT'   , 'PSLAT'   ,
X 'PSTR'    /

```

#### DATA SN3/

```

X 'T1'      , 'dv1snx'  , 'dv1sny'  , 'dv1snz'  , 'dp1'     ,
X 'dq1'     , 'dr1'     , 'dp2'     , 'dq2'     , 'dr2'     ,
X 'ddra2s2x' , 'ddra2s2y' , 'ddra2s2z' , 'v1snx'   , 'v1sny'   ,
X 'v1snz'   , 'p1'      , 'q1'      , 'r1'      , 'p2'      ,
X 'q2'      , 'r2'      , 'dra2s2x' , 'dra2s2y' , 'dra2s2z' ,
X 'dph1'    , 'dth1'    , 'dps1'    , 'dph2'    , 'dth2'    ,
X 'dps2'    , 'rlsnx'   , 'rlsny'   , 'rlsnz'   , 'ph1'     ,
X 'th1'     , 'ps1'     , 'ph2'     , 'th2'     , 'ps2'     ,
X 'ra2s2x'  , 'ra2s2y'  , 'ra2s2z'  , 'p2p'     , 'q2p'     ,
X 'fc1lx'   , 'fc1ly'   , 'fc1lz'   , 'fc1nx'   , 'fc1ny'   ,
X 'fc1nz'   , 'LC1'     , 'MC1'     , 'NC1'     , 'fallx'   ,
X 'fally'   , 'fallz'   , 'falnx'   , 'falny'   , 'fainz'   ,
X 'LA1'     , 'MA1'     , 'NA1'     , 'fa22x'   , 'fa22y'   ,
X 'fa22z'   , 'fa2nx'   , 'fa2ny'   , 'fa2nz'   , 'LA2'     ,
X 'MA2'     , 'NA2'     , 'fa2wx'   , 'fa2wy'   , 'fa2wz'   ,
X 'LA2W'    , 'MA2W'    , 'NA2W'    , 'v2snx'   , 'v2sny'   ,
X 'v2snz'   , 'va2s2x'  , 'va2s2y'  , 'va2s2z'  , 'ra2s1x'  ,
X 'ra2s1y'  , 'ra2s1z'  , 'rls2snx' , 'rls2sny' , 'rls2snz' ,
X 'da'      , 'db'      , 'dc'      , 'dp'      , 'RSAS'    ,
X 'PSAS'    , 'YSAS'    , 'DMIXA'   , 'DMIXB'   , 'DMIXC'   ,
X 'DMIXP'   , 'PSFWD'   , 'PSAFT'   , 'PSLAT'   , 'PSTR'    ,
X 'quad'    , 'alf2d'   , 'bet2d'   , 'dalf2d'  , 'dbet2d'  ,
X 'dang2d'  /

```

#### DATA SN4/

```

X 'T1'      , 'p1'      , 'q1'      , 'r1'      , 'da'      ,
X 'db'      , 'dc'      , 'dp'      , 'p2p'     , 'q2p'     ,
X 'RSAS'    , 'PSAS'    , 'YSAS'    , 'DMIXA'   , 'DMIXB'   ,
X 'DMIXC'   , 'DMIXP'   /

```

```

DATA SNW/
X 'T'      , 'alf'      , 'blf'      , 'vlx'      , 'vly'      ,
X 'vlz'    , 'vo'       , 'volx'     , 'volz'     , 'volz'     ,
X 'vpplx'  , 'vpplx'     , 'vpplx'    , 'vpplx'    , 'vpplx'    ,
X 'vppnz'  , 'vpp'       , 'vplx'     , 'vply'     , 'vply'     ,
X 'vpnx'   , 'vpny'      , 'vpnz'     , 'vp'       , 'phiw'     ,
X 'thew'   , 'xcw'       , 'ycw'      , 'rh2swx'   , 'rh2swy'   ,
X 'rh2swz' , 'rh2swox'   , 'rh2swoy'  , 'rh2swoz'  , 'thetap'   ,
X 'radius' , 'rad'       , 'height'   , 'vwz'      , 'vwt'      ,
X 'pst'    , 'vwvx'     , 'vwvy'     , 'vwvz'     , 'vw2x'     ,
X 'vw2y'   , 'vw2z'     , 'va2s2wx'  , 'va2s2wy'  , 'va2s2wz'  ,
X 'va2s2x' , 'va2s2y'   , 'va2s2z'   , /

```

```

DATA SNX/
X 'T'      , 'alf'      , 'blf'      , 'vo'       , 'vpp'      ,
X 'vp'     , 'phiw'     , 'thew'     , 'xcw'      , 'ycw'      ,
X 'rh2swox' , 'rh2swoy' , 'rh2swoz' , 'thetap'   , 'radius'   ,
X 'height' , 'vwz'      , 'pst'      , 'va2s2wx'  , 'va2s2wy'  ,
X 'va2s2wz' , /

```

C Added for automatic data collection: case name is read in from file

```
OPEN(9,FILE='fname.dat',FORM='unformatted',STATUS='old')
```

```
READ(9) NAME
```

C Without automatic data collection, file name is manually entered

```
c10 TYPE *, 'Enter name of input file (name.dat):'
```

```
c READ(5,11,ERR=10) NAME
```

```
NLAST = LASTCHR(NAME)
```

```
FIN = NAME(1:nlast)//".dat"
```

```
FOUT = NAME(1:nlast)//".out"
```

```
FXP = NAME(1:nlast)//".xp"
```

```
WRITE(6,11) FIN
```

```
11 FORMAT(A)
```

```
OPEN(1,FILE=FIN,FORM='unformatted',STATUS='old')
```

```
READ(1) NS, DT, ILOAD, IWAKE, ISWIRL, IPILOT, IDATA, NGAJFPS,
```

```
X TOW, FWT, W1, I1XX, I1YY, I1ZZ, I1XZ, AXIS, R1SN,
```

```
X PH1DEG, TH1DEG, PS1DEG, V1SN, PSVA, FA11, MA11, C
```

```
READ(1) CHFILE
```

```
OPEN(2,FILE=FOUT,FORM='formatted',STATUS='unknown')
```

```
WRITE(2,151) FOUT
```

```
IF (ILOAD.EQ.0) L = openW(UNIT,FXP,NCHAN1,SN1,'unc3')
```

```
IF (ILOAD.EQ.1) L = openW(UNIT,FXP,NCHAN2,SN2,'unc3')
```

```
IF ((ILOAD.EQ.2).AND.(IDATA.EQ.1))
```

```
X L = openW(UNIT,FXP,NCHAN3,SN3,'unc3')
```

```
IF ((ILOAD.EQ.2).AND.(IDATA.EQ.0))
```

```
X L = openW(UNIT,FXP,NCHAN4,SN4,'unc3')
```

```
IF (.NOT.L) STOP'openW'
```

```
IF (ILOAD.EQ.1) THEN
```

```
READ(1) LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ, R1SA1, R2S12,
```



```

X      DOQ, STRETCH, LCO, LC, KS, CS, QO, V1S1, FA1N, FA2N,
X      R1SN, PH2, TH2, PS2, PHC, THC, LC, (U(j),j=1,12)
      CABLES = ' '
      WRITE(2,101)
END IF

IF (ILOAD.EQ.2) THEN
      READ(1) LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ, STRETCH,
X      IAERSL, NC, R2SJ2, RA2PO2, R2P2SO2, R2P2S2, KS, CS,
X      R1SA1, R2PJ2, DOQ, GAMA, PSVA, ALF2D, BET2D, DETG, LCJO,
X      LCJ, TAUJ, FA1N, FC11, FC1N, MC11, FA22, FA2N, MA22, V1SN,
X      PH2DEG, TH2DEG, PS2DEG, RA2S2, VA2SN, R1S2SN, DYNAMIC
      CABLES = 'S'
      WRITE(2,102)
END IF
READ(1) NREC
RUNTIME = NREC * DT

IF (CHFILE.EQ.'sweep.dat') THEN
      IF (AXIS.EQ.1) THEN
            WRITE(2,103)
      ELSE IF (AXIS.EQ.2) THEN
            WRITE(2,104)
      ELSE IF (AXIS.EQ.3) THEN
            WRITE(2,105)
      ELSE
            WRITE(2,106)
      END IF
ELSE
      WRITE(2,107) CHFILE
END IF

IF (ILOAD.EQ.0) WRITE(2,100)

WRITE(2,152) NREC, DT, I1XX, I1YY, I1ZZ, I1XZ
WRITE(2,114) TOW, FWT, W1

IF (ILOAD.EQ.0) THEN
      WRITE(2,153) R1SN, PH1DEG, TH1DEG, PS1DEG, V1SN, PSVA,
X      FA11, MA11, C
ELSE
      WRITE(2,115) W1+W2
      WRITE(2,112) LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ
      IF (ILOAD.EQ.2) WRITE(2,113) RA2PO2, R2P2SO2, R2P2S2,
X      (R2PJ2(1,J),R2PJ2(2,J),R2PJ2(3,J),J=1,NC),R1S2SN

      IF (STRETCH.EQ.0) WRITE(2,116) CABLES
      IF (STRETCH.EQ.1) WRITE(2,117) CABLES

      IF ((IAERSL.EQ.0).OR.((IAERSL.EQ.1).AND.(DOQ.EQ.0))) THEN
            WRITE(2,118)
      ELSE
            IF (IAERSL.EQ.1) WRITE(2,119)
            IF (IAERSL.EQ.2) WRITE(2,120) DYNAMIC

```

```

        IF (IWAKE.EQ.0) WRITE(2,121)
        IF (IWAKE.EQ.1) THEN
            IF (ISWIRL.EQ.0) WRITE(2,122)
            IF (ISWIRL.EQ.1) WRITE(2,123)
        END IF
    END IF

END IF

IF (NGAJFPS.EQ.0) WRITE(2,124)
IF (NGAJFPS.EQ.1) WRITE(2,125)

IF (IPILOT.EQ.0) WRITE(2,126)
IF (IPILOT.EQ.1) WRITE(2,127)

IF (ILOAD.EQ.1) THEN
    WRITE(2,201) R1SA1, R2S12, LCO
    IF (STRETCH.EQ.1) WRITE(2,202) KS, CS
    WRITE(2,203) R1SN, PH1DEG, TH1DEG, PS1DEG, PH2, TH2, PS2,
X    PHC, THC, LC, V1S1, FA1N, FA2N, MA11, (Q(j),U(j),j=1,12)
END IF

IF (ILOAD.EQ.2) THEN
    WRITE(2,301) R1SA1
    IF (STRETCH.NE.0) WRITE(2,302) KS, CS, NC, DETG,
X    (TAUJ(j),LCJ(j),LCJO(j),j=1,NC)
    THT = ATAN(SQRT(FA2N(1)**2+FA2N(2)**2)/(W2+FA2N(3)))*RTD
    R1SN(3) = -R1SN(3)
    WRITE(2,305) R1SN, V1SN, PH1DEG, TH1DEG, PS1DEG,
X    PH2DEG, TH2DEG, PS2DEG, RA2S2, VA2SN, FA11, FA1N, MA11,
X    FC11, FC1N, MC11, FA22, FA2N, MA22, ALF2D, BET2D, THT, C
END IF

C  STORE XPLOT FILE
C  read time histories.  Convert angles to degrees, change signs
C  of y,z position coordinates for r1sn, r2sn, rsn

IF (ILOAD.EQ.0) THEN
    DO 511 i = 1,NREC-1
        READ(1,err=590) DAT1
        TIME = DAT1(1)
        DO 512 j = 1,12
            k = JRTD1(j)
512    DAT1(k) = DAT1(k)*RTD
        DO 513 j = 1,NCHAN1
513    DAT81(j) = DAT1(j)
511    CALL fWrite(UNIT, TIME, DAT81)
        CALL closeW(UNIT)
        WRITE(6,580) FOUT, RUNTIME, FXP
        STOP
END IF

IF (ILOAD.EQ.1) THEN
    DO 521 i = 1,NREC-1
        READ(1,err=590) DAT2

```

```

        TIME = DAT2(1)
        DAT2(14) = DAT2(14)/W1
        DO 522 j = 1,31
            k = JR2D2(j)
522      DAT2(k) = DAT2(k)*RTD
        DO 523 k = 1,2
            DAT2(35+k) = - DAT2(35+k)
            DAT2(47+k) = - DAT2(47+k)
523      DAT2(56+k) = - DAT2(56+k)
        DO 524 j = 1,NCHAN2
524      DAT82(j) = DAT2(j)
521      CALL fWrite(UNIT, TIME, DAT82)
        CALL closeW(UNIT)
        WRITE(6,580) FOUT, RUNTIME, FXP
    END IF
    IF ((ILOAD.EQ.2).AND.(IDATA.EQ.1)) THEN
        DO 531 i = 1,NREC-1
            READ(1,err=590) DAT3
            TIME = DAT3(1)
            DO 532 j = 1,26
                k = JR2D3(j)
532      DAT3(k) = DAT3(k)*RTD
            DAT3(34) = - DAT3(34)
C    correct load yaw angle to within +/- 180 deg for viewing in XPLOT
534      IF (DAT3(40).GT. 180.) DAT3(40) = DAT3(40) - 360
            IF (DAT3(40).LT.-180.) DAT3(40) = DAT3(40) + 360
            IF (DAT3(40).GT. 180.) GO TO 534
            IF (DAT3(40).LT.-180.) GO TO 534
        DO 533 j = 1,NCHAN3
533      DAT83(j) = DAT3(j)
531      CALL fWrite(UNIT, TIME, DAT83)
        CALL closeW(UNIT)
        WRITE(6,580) FOUT, RUNTIME, FXP
    END IF
    IF ((ILOAD.EQ.2).AND.(IDATA.EQ.0)) THEN
        DO 561 i = 1,NREC-1
            READ(1,err=590) DAT4
            TIME = DAT4(1)
            DO 562 j = 1,5
                k = JR2D4(j)
562      DAT4(k) = DAT4(k)*RTD
        DO 563 j = 1,NCHAN4
563      DAT84(j) = DAT4(j)
561      CALL fWrite(UNIT, TIME, DAT84)
        CALL closeW(UNIT)
        WRITE(6,580) FOUT, RUNTIME, FXP
    END IF
599  CONTINUE
    IF (IWAKE.EQ.1) THEN
        FIN = NAME(1:nlast)//".da2"
        FXP = NAME(1:nlast)//".wake.xp"
        WRITE(6,11) FIN
        OPEN(4,FILE=FIN,FORM='unformatted',STATUS='old')

```

```

        IF (IDATA.EQ.1) L = openW(UNIT,FXP,NCHANW,SNW,'unc3')
        IF (IDATA.EQ.0) L = openW(UNIT,FXP,NCHANX,SNX,'unc3')
        L = openW(UNIT,FXP,NCHANX,SNX,'unc3')
        IF (.NOT.L) STOP'openW'
        READ(4) junk
        DO 600 i = 1,NREC-1
        IF (IDATA.EQ.1) THEN
            READ(4,err=592) DATW
            TIME = DATW(1)
            DO 650 j = 1,NCHANW
            DAT8W(j) = DATW(j)
            CALL fWrite(UNIT, TIME, DAT8W)
        ELSE IF (IDATA.EQ.0) THEN
            READ(4,err=592) DATX
            TIME = DATX(1)
            DO 651 j = 1,NCHANX
            DAT8X(j) = DATX(j)
            CALL fWrite(UNIT, TIME, DAT8X)
        END IF
        600 CONTINUE
        CALL closeW(UNIT)
        WRITE(6,680) FXP
        END IF
        STOP

151 FORMAT('DATA OUTPUT FILE ',16A//)
100 FORMAT('NO LOAD SIMULATION')
101 FORMAT('SINGLE LOAD, SINGLE CABLE SIMULATION')
102 FORMAT('SINGLE LOAD, MULTI-CABLE SIMULATION')
103 FORMAT('COMPUTER GENERATED LATERAL INPUT SWEEP//')
104 FORMAT('COMPUTER GENERATED LONGITUDINAL INPUT SWEEP//')
105 FORMAT('COMPUTER GENERATED DIRECTIONAL INPUT SWEEP//')
106 FORMAT('COMPUTER GENERATED COLLECTIVE INPUT SWEEP//')
107 FORMAT('CONTROL INPUT FROM '40A//)
112 FORMAT(5X'LOAD DESCRIPTION : 'A/
X      5X'LOAD WEIGHT : 'F8.2,' LB'/
X      5X'LOAD INERTIA XX : 'F8.2,
X      3X'LOAD INERTIA YY : 'F8.2,' LB-FT-S**2'/
X      5X'LOAD INERTIA ZZ : 'F8.2,
X      3X'LOAD INERTIA XZ : 'F8.2,' LB-FT-S**2')
113 FORMAT(5X'RA2PO2 : '3F8.2/
X      5X'R2P2SO2 : '3F8.2/
X      5X'R2P2S2 : '3F8.2/
X      5X'R2PJ2 : '8(3F8.2/17X)/
X      5X'R1S2SN : '3F8.2)
114 FORMAT(5X'HELICOPTER TAKEOFF WEIGHT : 'F10.2,' LB'/
X      5X' FUEL WEIGHT : 'F10.2,' LB'/
X      5X' CURRENT WEIGHT : 'F10.2,' LB'/)
115 FORMAT(5X'TOTAL HELICOPTER AND LOAD : 'F10.2,' LB'/)
116 FORMAT(5X'INELASTIC CABLE',A)
117 FORMAT(5X'ELASTIC CABLE',A)
118 FORMAT(5X'NO LOAD AERODYNAMICS')
119 FORMAT(5X'DRAG ONLY AERODYNAMICS')
120 FORMAT(5X'CONEX STATIC AERODYNAMICS'/

```

```

X      5X'    DYNAMIC AERODYNAMIC COEFFICIENT = ',F4.1/)
121  FORMAT(5X'WAKE MODEL NOT SELECTED')
122  FORMAT(5X'WAKE MODEL WITHOUT SWIRL SELECTED')
123  FORMAT(5X'3-D WAKE MODEL SELECTED')
124  FORMAT(5X'FLIGHT PATH STABILIZATION (FPS) DISENGAGED')
125  FORMAT(5X'FLIGHT PATH STABILIZATION (FPS) ENGAGED')
126  FORMAT(5X'INPUT CONTROL HISTORY AUTOPILOT DISENGAGED')
127  FORMAT(5X'INPUT CONTROL HISTORY AUTOPILOT ENGAGED')

152  FORMAT(5X'NREC, DT      :    'I8,F10.2/
X      5X'HC INERTIA XX      :    'F8.2,
X      3X'HC INERTIA YY      :    'F8.2,' LB-FT-S**2',/
X      5X'HC INERTIA ZZ      :    'F8.2,
X      3X'HC INERTIA XZ      :    'F8.2,' LB-FT-S**2',//)
153  FORMAT('TRIM'/
X      5X'R1SN (x,y,z)        ', 3F10.1/
X      5X'PH1, TH1, PS1 (DEG) ', 3F10.2/
X      5X'V1SN (x,y,z)        ', 3F10.2/
X      5X'PSVA                ', 1F10.2/
X      5X'FA11 (x,y,z)        ', 3F10.1/
X      5X'MA11 (l,m,n)        ', 3F10.1/
X      5X'XAAD, XBAD, XCAD, XPAD', 4F10.2/)

201  FORMAT(5X'R1SA1          ' 3f10.2/
X      5X'R2S12              ' 3f10.2/
X      5X'LCO                ' f10.4/)
202  FORMAT(5X'KS, CS        ' 2f10.1/)
203  FORMAT('TRIM'/
X      5X'R1SN                '3F9.1/
X      5X'PH1, TH1, PS1 (deg)', 3F9.2/
X      5X'PH2, TH2, PS2 (deg)', 3F9.2/
X      5X'PHC, THC, LC        ', 2F9.2, F9.4/
X      5X'VREFN               ', 3F9.2/
X      5X'FA1N                ', 3F9.1/
X      5X'FA2N                ', 3F9.1/
X      5X'MA11                ', 3F9.1//
X      'INITIAL STATES, INCLUDING INITIAL OFFSETS FROM TRIM'/
X      5X'          q - ft,deg      u - fps,rps '/
X      5X'R1SNX', f15.6, 7x, f10.6/
X      5X'R1SNY', f15.6, 7x, f10.6/
X      5X'R1SNZ', f15.6, 7x, f10.6/
X      5X'PH1  ', f15.6, 7x, f10.6/
X      5X'TH1  ', f15.6, 7x, f10.6/
X      5X'PS1  ', f15.6, 7x, f10.6/
X      5X'PH2  ', f15.6, 7x, f10.6/
X      5X'TH2  ', f15.6, 7x, f10.6/
X      5X'PS2  ', f15.6, 7x, f10.6/
X      5X'PHC  ', f15.6, 7x, f10.6/
X      5X'THC  ', f15.6, 7x, f10.6/
X      5X'LC   ', f15.6, 7x, f10.6// )

301  FORMAT(5X'R1SA1          '3f8.2)
302  FORMAT(5X'KS, CS, NC, DETG ' 2F8.1,I4,F8.4/
X      5X'TAUJ, LCJ, LCJO     ' 8(F8.1,3X,2F10.4/25X))

```

```

305  FORMAT('TRIM'/
X      5X'R1SN              ', 3F9.1/
X      5X'V1SN              ', 3F9.2/
X      5X'PH1, TH1, PS1 (deg)', 3F9.2/
X      5X'PH2, TH2, PS2 (deg)', 3F9.2/
X      5X'RA2S2              ', 3F9.2/
X      5X'VA2SN              ', 3F9.2/
X      5X'FA11, FA1N         ', 6F9.1/
X      5X'MA11              ', 3F9.1/
X      5X'FC11, FC1N         ', 6F9.1/
X      5X'MC11              ', 3F9.1/
X      5X'FA22, FA2N         ', 6F9.1/
X      5X'MA22              ', 3F9.1/
X      5X'ALF2D, BET2D (deg)', 2F9.2/
X      5X'TRAIL ANGLE (deg)', F10.2/
X      5X'da, db, dc, dp    ', 4F9.3)

580  FORMAT(/
X'    RUN INFORMATION AND TRIM DATA WRITTEN TO ',A/
X'    ',1F7.2,' SEC SIMULATION RECORD WRITTEN TO ',A/)

590  TYPE *, 'error in reading data array'
WRITE(6,591) I
591  FORMAT('stopped at record number',I4)
CALL closeW(unit)
GO TO 599
592  TYPE *, 'error in reading data array'
WRITE(6,593) I
593  FORMAT('stopped at record number',I4)
CALL closeW(unit)

680  FORMAT(/
X'    WAKE INFORMATION WRITTEN TO ',A/)

END

```

## B.5. SUBROUTINE `ghsl_init`

*Functions:* Initialization subroutine for Gen Hel/SL. Prints load configuration and slung load run parameters to standard output during non-real-time program operation.

*Inputs:* `ghsl.dat` namelists:

HCDATA: TOW, XMOMTO, FWT, FWMX, AIRSPEED  
SLRUN: CHFILE, AXIS, DATAFILE, STRETCH, NSTORE,  
IAERSL, ILOAD, IPILOT, IWAKE, ISWIRL,  
IDATA, DYNAMIC  
SLDATA: LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ, KS,  
CS, LCO, RA2PO2, R2P2SO2, R2P2S2, NC,  
R2PJ2, R2S12, DOQ, DELPS20

*Outputs:* W1, R1SA1, FSCG, WLCG, BLCG, initialized namelist variables.

*Called By:* `bhawk_nrt_exec`

*Calls:* Reads data file `ghsl.dat` found in the Genhel/batch directory.

*Comments:* DATA block in subroutine initializes all RLOAD variables to zero, prior to reading in namelists variables. Also, DATA block includes the locations of the hub, hook, and fuel tank center of gravity (fuselage station, buttline, waterline coordinates). The variables for helicopter weight and trim airspeed are also found in `bhawk.dat`, but the values in `ghsl.dat` take effect as they are read in last.

### *Program Listing:*

```
C ghsl_init.f ----- 31 AUG 98 Peter Tyson
C Initialization for GenHel/Slung Load Simulation.
C Reads input data from 'ghsl.dat' which resides in the GenHel/batch
C directory. Load Configuration and Slung Load run parameters are
C appended to the screen during nrt program run (only for SLSC or
C SLMC)
C Subroutine is called by BHAWK_NRT_EXEC immediately following call
C for BHAWK_NRT_INIT.
```

```
SUBROUTINE GHSL_INIT
INCLUDE 'slvars.cmn'
```

```
REAL FSHK, BLHK, WLHK, FSHB, BLHB, WLHB, FSCGTNK, FSWI, FWMX,
X    ESBLCG, ESWLCG, AIRSPEED
CHARACTER CABLES
EQUIVALENCE (A(238), AIRSPEED)
```

```

DATA
X   RLOAD                /500*0.0/
X   DELPS2O              /0.0/,
X   ALF2D, BET2D         /0.0, 0.0/,
X   NREC                 /0/
X   ICUNIT, IOUT         /1, 6/
X   FSHK, WLHK, BLHK    /352.6, 195.5, 0./,
X   FSHB, WLHB, BLHB    /341.2, 315.0, 0./,
X   FSCGTNK, FSWI       /420.8, 6.8/,
X   ESBLCG, ESWLCG      /0.0, 247.2/

NAMELIST /HCDATA/ TOW, XMOMTO, FWT, FWMX, AIRSPEED
NAMELIST /SLRUN/ CHFILE, AXIS, DATAFILE, STRETCH, NSTORE,
X   IAERSL, ILOAD, IPILOT, IWAKE, ISWIRL, IDATA, DYNAMIC
NAMELIST /SLDATA/ LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ, KS,
X   CS, LCO, RA2PO2, R2P2SO2, R2P2S2, NC, R2PJ2, R2S12, DOQ,
X   DELPS2O
OPEN(ICUNIT,FILE='ghsl.dat',STATUS='OLD',READONLY,ERR=14)
READ(ICUNIT,HCDATA)
REWIND ICUNIT
READ(ICUNIT,SLRUN)
REWIND ICUNIT
READ(ICUNIT,SLDATA)
CLOSE(ICUNIT)

XMOMI  = XMOMTO - FWMX*FSCGTNK
ZMOMI  = TOW*ESWLCG - (204.75 + .09116*FWMX/FSWI)*FWMX
W1      = TOW - FWMX + FWT
FSCG    = (XMOMI + FWT*FSCGTNK)/W1
WLCG    = (ZMOMI + (204.75 + .09116*FWT/FSWI)*FWT)/W1
BLCG    = ESBLCG

C--- position vector, HC c.g. to hook, HC body axes ---C
R1SA1(1) = (FSCG - FSHK)/12.
R1SA1(2) = (-BLCG + BLHK)/12.
R1SA1(3) = (WLCG - WLHK)/12.

IF (CHFILE.NE.'sweep.dat') AXIS = 1
IF (ILOAD.EQ.0) WRITE(IOUT,100)
IF (ILOAD.EQ.1) THEN
  WRITE(IOUT,101)
  CABLES = ' '
END IF
IF (ILOAD.EQ.2) THEN
  WRITE(IOUT,102)
  CABLES = 'S'
END IF
IF (ILOAD.NE.0) THEN
  WRITE(IOUT,103) LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ, RA2PO2,
X   R2P2SO2, R2P2S2, (R2PJ2(1,J),R2PJ2(2,J),R2PJ2(3,J),J=1,NC)
  WRITE(IOUT,104) TOW, W1, FWT
  WRITE(IOUT,105) KS, CS, DT
  IF (STRETCH.EQ.0) WRITE(IOUT,106) CABLES
  IF (STRETCH.EQ.1) WRITE(IOUT,107) CABLES
  IF ((IAERSL.EQ.1).AND.(DOQ.EQ.0.)) WRITE (IOUT,108)

```



```

        IF (IAERSL.EQ.1) WRITE(IOUT,109)
        IF (IAERSL.EQ.2) WRITE(IOUT,110)
        IF (IWAKE.EQ.0) WRITE(IOUT,111)
        IF (IWAKE.EQ.1) THEN
            IF (ISWIRL.EQ.0) WRITE(IOUT,112)
            IF (ISWIRL.EQ.1) WRITE(IOUT,113)
        END IF
    END IF
    IF (IPILOT.EQ.0) WRITE(IOUT,114)
    IF (IPILOT.EQ.1) WRITE(IOUT,115)
    IF (CHFILE.EQ.'sweep.dat') THEN
        IF (AXIS.EQ.1) WRITE(IOUT,116)
        IF (AXIS.EQ.2) WRITE(IOUT,117)
        IF (AXIS.EQ.3) WRITE(IOUT,118)
        IF (AXIS.EQ.4) WRITE(IOUT,119)
    END IF
    RETURN

100 FORMAT(5X'NO LOAD SIMULATION')
101 FORMAT(5X'SINGLE LOAD, SINGLE CABLE SIMULATION')
102 FORMAT(5X'SINGLE LOAD, MULTI-CABLE SIMULATION')
103 FORMAT(//
    X'                                L O A D   C O N F I G U R A T I O N'//
    X5X'LOAD DESCRIPTION : 'A//
    X5X'LOAD WEIGHT      : 'F8.2//
    X5X'LOAD INERTIA XX  : 'F8.2,
    X'  LOAD INERTIA YY  : 'F8.2,'    LB-FT-S**2',/
    X5X'LOAD INERTIA ZZ  : 'F8.2,
    X'  LOAD INERTIA XZ  : 'F8.2,'    LB-FT-S**2',//
    X5X'RA2PO2          : '3F8.2/
    X5X'R2P2SO2         : '3F8.2/
    X5X'R2P2S2          : '3F8.2//
    X5X'R2PJ2           : '/8(10X,3F8.2//)
104 FORMAT(5X'T/O WT, CURRENT WT, FUEL WT: '3F10.2//)
105 FORMAT(5X'KS, CS, DT : '3F8.2//)
106 FORMAT(5X'INELASTIC CABLE',A)
107 FORMAT(5X'ELASTIC CABLE',A)
108 FORMAT(5X'NO AERODYNAMICS')
109 FORMAT(5X'DRAG ONLY AERODYNAMICS')
110 FORMAT(5X'CONEX AERODYNAMICS')
111 FORMAT(5X'WAKE MODEL NOT SELECTED')
112 FORMAT(5X'AXIAL WAKE MODEL SELECTED')
113 FORMAT(5X'3-D WAKE MODEL SELECTED')
114 FORMAT(5X'INPUT CONTROL HISTORY AUTOPILOT DISENGAGED')
115 FORMAT(5X'INPUT CONTROL HISTORY AUTOPILOT ENGAGED')
116 FORMAT(5X'LATERAL AXIS COMPUTER GENERATED FREQUENCY SWEEP')
117 FORMAT(5X'LONGITUDINAL AXIS COMPUTER GENERATED FREQUENCY SWEEP')
118 FORMAT(5X'COLLECTIVE AXIS COMPUTER GENERATED FREQUENCY SWEEP')
119 FORMAT(5X'DIRECTIONAL AXIS COMPUTER GENERATED FREQUENCY SWEEP')
14  WRITE(6,12)
12  FORMAT('!!! ERROR OPENING INPUT FILE ghs1.dat !!!')
    STOP
    END

```

## B.6. SUBROUTINE **ghslmc**

*Functions:* Compute the forces and moments acting on the helicopter at the cargo hook due to the presence of the load for the single lift, multi-cable suspension configuration.

*Inputs:* FA11, MA11 (aerodynamic forces/moments acting on helicopter, in helicopter body axes, from ftotal)  
Slung load variables contained in slvars.cmn

*Outputs:* FC11, MC11 (forces/moments at hook, helicopter body axes)

*Called By:* ftotal

*Calls:* conexaero, ghslmc\_ic, wake

*Comments:* Originally written by Mr. Luigi Cicolani as slmc and used in the SL\_DRIVER simulation program, ghslmc was modified to conform to Gen Hel variable names, to incorporate conexaero and wake, to utilize the variables contained in slvars.cmn, and to accommodate Gen Hel's 2-component rigid body (Main Rotor/Helicopter) dynamic system structure.

### *Program Listing:*

C ghslmc.f, originally slmc.f ..... started 1 july 96  
C single lift, multi-cable suspension.  
C modified for use with Gen Hel UH-60A model

```
SUBROUTINE GHSLMC
```

```
INCLUDE 'slvars.cmn'
```

```
REAL T1N(3,3), TN1(3,3), A22(3,3), A23(3,3), A22J1I(3,3),  
X A23J2I(3,3), RJR1(3,3), RJR2(3,3),  
X TV12(3), TV22(3), TV2N(3), TV3N(3), TV4N(3), TV5N(3),  
X TV6N(3), TV7N(3), TV8N(3), CA11(3), DAU2N(3), JOM11(3),  
X JOM22(3), X11(3), X22(3), FO1N(3), FO2N(3), MO11(3),  
X MO22(3), HTDIFO(3), VSON(3), V1SN(3), V1S2SN(3), COR1N(3),  
X COR2N(3), DRA2SN(3), RSON(3), FC12(3),  
X SF1N(3), SF2N(3), SM11(3), SM22(3), SUMM11(3), SUMM22(3),  
X DDRA2SN(3), DV1S2SN(3), DVS2N(3), ADU2N(3), OSSFN(3),  
X OISSFN(3), T1T2(3,3)  
REAL  
X II1XX, II1YY, II1ZZ, II1XZ, II2XX, II2YY, II2ZZ, II2XZ,  
X M1, M2, MU12, M1PM2, M2OM, NQSL,  
X ODU(12), ODQ(12), DLCJ(8), RAJ2(3,8), KCJ2(3,8),  
X DELQIC(12), C(4), KCN(3), KCPN(3), KCPH(3), KCP1(3),  
X RA2SN(3), KCXKCPN(3), KCXK2N(3), K2N(3), KN2(3),  
X DNRA2S2(3), DNRA2SN(3), DV2S2(3), DVAN(3),  
X OMXR22(3), DV2SP2(3), DV2SPN(3), OMXDR22(3), OMXDR2N(3),
```

```

X      DV2SSN(3), DV2SCHKN(3), DV2SPON(3), OM21(3),
X      GMA2SN(3), GMA2S2(3)

EQUIVALENCE (U(1)      , V1SN(1) )
EQUIVALENCE (Q(6)      , PS1 )
EQUIVALENCE (TN2(1,3), K2N(1) )
EQUIVALENCE (T2N(1,3), KN2(1) )

C      COMPUTE DEPENDENT PARAMETERS AND LOAD-SUSPENSION
C      POSITION COORDINATES FOR STATIC EQUILIBRIUM.

C--- idle mode (not used)
      IF (IMODE.EQ.0) RETURN
      DTO2 = .5*DT

C--- initial conditions
C current helicopter mass and inertia from STRIKE
      M1      = W1/G
      R1      = I1XZ/(I1XX*I1ZZ)
      R2      = 1 - R1*I1XZ
      II1XX   = 1/(I1XX*R2)
      II1XZ   = R1/R2
      II1YY   = 1/I1YY
      II1ZZ   = 1/(I1ZZ*R2)

C load parameters
      M2      = W2/G
      R1      = I2XZ/(I2XX*I2ZZ)
      R2      = 1 - R1*I2XZ
      II2XX   = 1/(I2XX*R2)
      II2XZ   = R1/R2
      II2YY   = 1/I2YY
      II2ZZ   = 1/(I2ZZ*R2)

C compute derived mass parameters
      MU12    = 1/M1 + 1/M2
      M1PM2   = M1 + M2
      M2OM    = M2/M1PM2

C--- initial conditions (trim) mode
C load-suspension states, add load yaw offset for yaw stability tests
      IF (IMODE.LT.0) THEN
        CALL GHSLMC_IC
c      PS2     = PS2 + DELPS20
        DO 10 I = 7,12
          U(I)   = 0.
10      DQ(I)   = 0.
C initial yaw offset changed to initial load yaw angular rate
      OM22(3) = DELPS20/R2D

C NP = data rate counter, store a record every NSTORE cycles
C NS counts records stored
      NP      = NSTORE
      NS      = 0

```

```

      DTO2 = 0.

C  magnetic dip angle at moffett
      DIP = 61.25/R2D
      SDIP = SIN(DIP)
      CDIP = COS(DIP)
      END IF
C***** OPERATE CODE *****

C*** SEC 100: read in HC states and position kinematics

C  read in HC states from STRIKE to q, dq, u, T1N
      PH1 = A( 4)
      TH1 = A( 5)
      PS1 = A( 6)
      R1SN(1) = A(106)
      R1SN(2) = A(107)
      R1SN(3) = -A(176)
      DO 102 I = 1,3
      V1SN(I) = A(63 + I)
      DQ(I) = V1SN(I)
      OM11(I) = A(36 + I)
102  DQ(3+I) = A( 6 + I)
      DO 101 I = 1,3
      DO 101 J = 1,3
      T1N(I,J) = A(15+I+(J-1)*3)
101  TN1(J,I) = A(15+I+(J-1)*3)

C  position kinematics
      SPH1 = SIN(PH1)
      CPH1 = COS(PH1)
      STH1 = SIN(TH1)
      CTH1 = COS(TH1)
      SPH2 = SIN(PH2)
      CPH2 = COS(PH2)
      STH2 = SIN(TH2)
      CTH2 = COS(TH2)
      SPS2 = SIN(PS2)
      CPS2 = COS(PS2)
      T2N(1,1) = CPS2*CTH2
      T2N(1,2) = CTH2*SPS2
      T2N(1,3) = -STH2
      T2N(2,1) = CPS2*SPH2*STH2-CPH2*SPS2
      T2N(2,2) = SPH2*SPS2*STH2+CPH2*CPS2
      T2N(2,3) = CTH2*SPH2
      T2N(3,1) = CPH2*CPS2*STH2+SPH2*SPS2
      T2N(3,2) = CPH2*SPS2*STH2-CPS2*SPH2
      T2N(3,3) = CPH2*CTH2
      TN2(1,1) = T2N(1,1)
      TN2(1,2) = T2N(2,1)
      TN2(1,3) = T2N(3,1)
      TN2(2,1) = T2N(1,2)
      TN2(2,2) = T2N(2,2)
      TN2(2,3) = T2N(3,2)

```

```

TN2(3,1) = T2N(1,3)
TN2(3,2) = T2N(2,3)
TN2(3,3) = T2N(3,3)

```

C\*\*\* SEC. 200. LOAD AERODYNAMICS.

```

C  load air velocity: v2sn = v1sn - TN1*S(r1sa1)*om11 -
C                      TN2*S(ra2s2)*om22 + Tn2*dra2s2
C                      or v2sn = v1sn + A22*om11 + A23*om22 + dra2sn
A22(1,1) = TN1(1,3)*R1SA1(2) - TN1(1,2)*R1SA1(3)
A22(1,2) = TN1(1,1)*R1SA1(3) - R1SA1(1)*TN1(1,3)
A22(1,3) = R1SA1(1)*TN1(1,2) - TN1(1,1)*R1SA1(2)
A22(2,1) = R1SA1(2)*TN1(2,3) - TN1(2,2)*R1SA1(3)
A22(2,2) = TN1(2,1)*R1SA1(3) - R1SA1(1)*TN1(2,3)
A22(2,3) = R1SA1(1)*TN1(2,2) - R1SA1(2)*TN1(2,1)
A22(3,1) = R1SA1(2)*TN1(3,3) - R1SA1(3)*TN1(3,2)
A22(3,2) = R1SA1(3)*TN1(3,1) - R1SA1(1)*TN1(3,3)
A22(3,3) = R1SA1(1)*TN1(3,2) - R1SA1(2)*TN1(3,1)
A23(1,1) = TN2(1,3)*RA2S2(2) - TN2(1,2)*RA2S2(3)
A23(1,2) = TN2(1,1)*RA2S2(3) - RA2S2(1)*TN2(1,3)
A23(1,3) = RA2S2(1)*TN2(1,2) - TN2(1,1)*RA2S2(2)
A23(2,1) = RA2S2(2)*TN2(2,3) - TN2(2,2)*RA2S2(3)
A23(2,2) = TN2(2,1)*RA2S2(3) - RA2S2(1)*TN2(2,3)
A23(2,3) = RA2S2(1)*TN2(2,2) - RA2S2(2)*TN2(2,1)
A23(3,1) = RA2S2(2)*TN2(3,3) - RA2S2(3)*TN2(3,2)
A23(3,2) = RA2S2(3)*TN2(3,1) - RA2S2(1)*TN2(3,3)
A23(3,3) = RA2S2(1)*TN2(3,2) - RA2S2(2)*TN2(3,1)
COR1N(1) = A22(1,3)*OM11(3)+A22(1,2)*OM11(2)+OM11(1)*A22(1,1)
COR1N(2) = A22(2,3)*OM11(3)+OM11(2)*A22(2,2)+OM11(1)*A22(2,1)
COR1N(3) = OM11(3)*A22(3,3)+OM11(2)*A22(3,2)+OM11(1)*A22(3,1)
COR2N(1) = A23(1,3)*OM22(3)+A23(1,2)*OM22(2)+OM22(1)*A23(1,1)
COR2N(2) = A23(2,3)*OM22(3)+OM22(2)*A23(2,2)+OM22(1)*A23(2,1)
COR2N(3) = OM22(3)*A23(3,3)+OM22(2)*A23(3,2)+OM22(1)*A23(3,1)
DRA2SN(1) = TN2(1,3)*DRA2S2(3)+TN2(1,2)*DRA2S2(2)+
x          DRA2S2(1)*TN2(1,1)
DRA2SN(2) = TN2(2,3)*DRA2S2(3)+DRA2S2(2)*TN2(2,2)+
x          DRA2S2(1)*TN2(2,1)
DRA2SN(3) = DRA2S2(3)*TN2(3,3)+DRA2S2(2)*TN2(3,2)+
x          DRA2S2(1)*TN2(3,1)
V1S2SN(1) = DRA2SN(1) + COR2N(1) + COR1N(1)
V1S2SN(2) = DRA2SN(2) + COR2N(2) + COR1N(2)
V1S2SN(3) = DRA2SN(3) + COR2N(3) + COR1N(3)
V2SN(1) = V1SN(1) + V1S2SN(1)
V2SN(2) = V1SN(2) + V1S2SN(2)
V2SN(3) = V1SN(3) + V1S2SN(3)
VA2SN(1) = V2SN(1) - WN(1)
VA2SN(2) = V2SN(2) - WN(2)
VA2SN(3) = V2SN(3) - WN(3)
VA2S2(1) = T2N(1,3)*VA2SN(3)+T2N(1,2)*VA2SN(2)+VA2SN(1)*T2N(1,1)
VA2S2(2) = T2N(2,3)*VA2SN(3)+VA2SN(2)*T2N(2,2)+VA2SN(1)*T2N(2,1)
VA2S2(3) = VA2SN(3)*T2N(3,3)+VA2SN(2)*T2N(3,2)+VA2SN(1)*T2N(3,1)

```

C\*\*\*\*\* LOAD AERODYNAMICS \*\*\*\*\*

C Load aero: option IAERSL = 0: no aero, 1: drag only, 2: CONEX aero

```

DO 110 I = 1,3
FA22(I) = 0.
110 MA22(I) = 0.
IF (IWAKE.EQ.1) CALL WAKE
IF (IAERSL.EQ.1) THEN
  VA2 = SQRT(VA2S2(1)**2 + VA2S2(2)**2 + VA2S2(3)**2)
  DOV = DOQ*RHO*VA2/2.
  FA22(1) = -DOV*VA2S2(1)
  FA22(2) = -DOV*VA2S2(2)
  FA22(3) = -DOV*VA2S2(3)
ELSE IF (IAERSL.EQ.2) THEN
  CALL CONEXAERO
END IF

C*** SEC 300. fo = fg + fa - X - D dA u

TV12(1) = OM22(2)*RA2S2(3) - RA2S2(2)*OM22(3) + 2.*DRA2S2(1)
TV12(2) = -OM22(1)*RA2S2(3) + RA2S2(1)*OM22(3) + 2.*DRA2S2(2)
TV12(3) = OM22(1)*RA2S2(2) - RA2S2(1)*OM22(2) + 2.*DRA2S2(3)
TV22(1) = OM22(2)*TV12(3) - TV12(2)*OM22(3)
TV22(2) = -OM22(1)*TV12(3) + TV12(1)*OM22(3)
TV22(3) = OM22(1)*TV12(2) - TV12(1)*OM22(2)
TV2N(1) = TN2(1,3)*TV22(3)+TN2(1,2)*TV22(2)+TV22(1)*TN2(1,1)
TV2N(2) = TN2(2,3)*TV22(3)+TV22(2)*TN2(2,2)+TV22(1)*TN2(2,1)
TV2N(3) = TV22(3)*TN2(3,3)+TV22(2)*TN2(3,2)+TV22(1)*TN2(3,1)
CA11(1) = OM11(2)*(OM11(1)*R1SA1(2)-R1SA1(1)*OM11(2))-
x OM11(3)*(R1SA1(1)*OM11(3)-OM11(1)*R1SA1(3))
CA11(2) = OM11(3)*(OM11(2)*R1SA1(3)-R1SA1(2)*OM11(3))-
x OM11(1)*(OM11(1)*R1SA1(2)-R1SA1(1)*OM11(2))
CA11(3) = OM11(1)*(R1SA1(1)*OM11(3)-OM11(1)*R1SA1(3))-
x OM11(2)*(OM11(2)*R1SA1(3)-R1SA1(2)*OM11(3))
DAU2N(1) = TN1(1,3)*CA11(3)+TN1(1,2)*CA11(2)+
x CA11(1)*TN1(1,1)+TV2N(1)
DAU2N(2) = TN1(2,3)*CA11(3)+CA11(2)*TN1(2,2)+
x CA11(1)*TN1(2,1)+TV2N(2)
DAU2N(3) = CA11(3)*TN1(3,3)+CA11(2)*TN1(3,2)+
x CA11(1)*TN1(3,1)+TV2N(3)
JOM11(1) = OM11(1)*I1XX-OM11(3)*I1XZ
JOM11(2) = OM11(2)*I1YY
JOM11(3) = OM11(3)*I1ZZ-OM11(1)*I1XZ
JOM22(1) = OM22(1)*I2XX-OM22(3)*I2XZ
JOM22(2) = OM22(2)*I2YY
JOM22(3) = OM22(3)*I2ZZ-OM22(1)*I2XZ
X11(1) = OM11(2)*JOM11(3) - JOM11(2)*OM11(3)
X11(2) = JOM11(1)*OM11(3) - OM11(1)*JOM11(3)
X11(3) = OM11(1)*JOM11(2) - JOM11(1)*OM11(2)
X22(1) = OM22(2)*JOM22(3) - JOM22(2)*OM22(3)
X22(2) = JOM22(1)*OM22(3) - OM22(1)*JOM22(3)
X22(3) = OM22(1)*JOM22(2) - JOM22(1)*OM22(2)
FA1N(1) = TN1(1,3)*FA11(3)+TN1(1,2)*FA11(2)+FA11(1)*TN1(1,1)
FA1N(2) = TN1(2,3)*FA11(3)+FA11(2)*TN1(2,2)+FA11(1)*TN1(2,1)
FA1N(3) = FA11(3)*TN1(3,3)+FA11(2)*TN1(3,2)+FA11(1)*TN1(3,1)

```

C add in the weight of the rotor system to the total aerodynamic

C force on the system because GENHEL treats the helo/rotor as a  
 C two body system, but the ghslmc logic requires entire helo as one  
 C body and the load as the second.

```

    FA1N(3) = FA1N(3) - WBLADE*NBS
    FA11(1) = T1N(1,3)*FA1N(3)+T1N(1,2)*FA1N(2)+FA1N(1)*T1N(1,1)
    FA11(2) = T1N(2,3)*FA1N(3)+FA1N(2)*T1N(2,2)+FA1N(1)*T1N(2,1)
    FA11(3) = FA1N(3)*T1N(3,3)+FA1N(2)*T1N(3,2)+FA1N(1)*T1N(3,1)

    FA2N(1) = TN2(1,3)*FA22(3)+TN2(1,2)*FA22(2)+FA22(1)*TN2(1,1)
    FA2N(2) = TN2(2,3)*FA22(3)+FA22(2)*TN2(2,2)+FA22(1)*TN2(2,1)
    FA2N(3) = FA22(3)*TN2(3,3)+FA22(2)*TN2(3,2)+FA22(1)*TN2(3,1)
    FO1N(1) = FA1N(1)
    FO1N(2) = FA1N(2)
    FO1N(3) = W1 + FA1N(3)
    FO2N(1) = FA2N(1) - M2*DAU2N(1)
    FO2N(2) = FA2N(2) - M2*DAU2N(2)
    FO2N(3) = W2 + FA2N(3) - M2*DAU2N(3)
    MO11(1) = MA11(1) - X11(1)
    MO11(2) = MA11(2) - X11(2)
    MO11(3) = MA11(3) - X11(3)
    MO22(1) = MA22(1) - X22(1)
    MO22(2) = MA22(2) - X22(2)
    MO22(3) = MA22(3) - X22(3)

```

C\*\*\* SEC 400: SUSPENSION FORCES ON HC AND LOAD      fc = H fc1n

IF (STRETCH.EQ.0) THEN

```

C inelastic cable; fc1n = -[HT*DI*H]^(-1)*HT*DI*fo
    A22J1I(1,1) = A22(1,3)*II1XZ+A22(1,1)*II1XX
    A22J1I(1,2) = A22(1,2)*II1YY
    A22J1I(1,3) = A22(1,3)*II1ZZ+A22(1,1)*II1XZ
    A22J1I(2,1) = A22(2,3)*II1XZ+A22(2,1)*II1XX
    A22J1I(2,2) = A22(2,2)*II1YY
    A22J1I(2,3) = A22(2,3)*II1ZZ+A22(2,1)*II1XZ
    A22J1I(3,1) = A22(3,3)*II1XZ+A22(3,1)*II1XX
    A22J1I(3,2) = A22(3,2)*II1YY
    A22J1I(3,3) = A22(3,3)*II1ZZ+A22(3,1)*II1XZ
    A23J2I(1,1) = A23(1,3)*II2XZ+A23(1,1)*II2XX
    A23J2I(1,2) = A23(1,2)*II2YY
    A23J2I(1,3) = A23(1,3)*II2ZZ+A23(1,1)*II2XZ
    A23J2I(2,1) = A23(2,3)*II2XZ+A23(2,1)*II2XX
    A23J2I(2,2) = A23(2,2)*II2YY
    A23J2I(2,3) = A23(2,3)*II2ZZ+A23(2,1)*II2XZ
    A23J2I(3,1) = A23(3,3)*II2XZ+A23(3,1)*II2XX
    A23J2I(3,2) = A23(3,2)*II2YY
    A23J2I(3,3) = A23(3,3)*II2ZZ+A23(3,1)*II2XZ
    TV3N(1) = A22J1I(1,3)*MO11(3)+A22J1I(1,2)*MO11(2)+
x      MO11(1)*A22J1I(1,1)
    TV3N(2) = A22J1I(2,3)*MO11(3)+MO11(2)*A22J1I(2,2)+
x      MO11(1)*A22J1I(2,1)
    TV3N(3) = MO11(3)*A22J1I(3,3)+MO11(2)*A22J1I(3,2)+
x      MO11(1)*A22J1I(3,1)
    TV4N(1) = A23J2I(1,3)*MO22(3)+A23J2I(1,2)*MO22(2)+

```

```

x      MO22(1)*A23J2I(1,1)
TV4N(2) = A23J2I(2,3)*MO22(3)+MO22(2)*A23J2I(2,2)+
x      MO22(1)*A23J2I(2,1)
TV4N(3) = MO22(3)*A23J2I(3,3)+MO22(2)*A23J2I(3,2)+
x      MO22(1)*A23J2I(3,1)
HTDIFO(1) = -FO2N(1)/M2+FO1N(1)/M1+TV4N(1)+TV3N(1)
HTDIFO(2) = -FO2N(2)/M2+FO1N(2)/M1+TV4N(2)+TV3N(2)
HTDIFO(3) = -FO2N(3)/M2+FO1N(3)/M1+TV4N(3)+TV3N(3)
RJR1(1,1) = A22(1,3)*A22J1I(1,3)+A22(1,2)*A22J1I(1,2)+
x      A22(1,1)*A22J1I(1,1)
RJR1(1,2) = A22J1I(1,3)*A22(2,3)+A22J1I(1,2)*A22(2,2)+
x      A22J1I(1,1)*A22(2,1)
RJR1(1,3) = A22J1I(1,3)*A22(3,3)+A22J1I(1,2)*A22(3,2)+
x      A22J1I(1,1)*A22(3,1)
RJR1(2,2) = A22(2,3)*A22J1I(2,3)+A22(2,2)*A22J1I(2,2)+
x      A22(2,1)*A22J1I(2,1)
RJR1(2,3) = A22J1I(2,3)*A22(3,3)+A22J1I(2,2)*A22(3,2)+
x      A22J1I(2,1)*A22(3,1)
RJR1(3,3) = A22(3,3)*A22J1I(3,3)+A22(3,2)*A22J1I(3,2)+
x      A22(3,1)*A22J1I(3,1)
RJR2(1,1) = A23(1,3)*A23J2I(1,3)+A23(1,2)*A23J2I(1,2)+
x      A23(1,1)*A23J2I(1,1)
RJR2(1,2) = A23J2I(1,3)*A23(2,3)+A23J2I(1,2)*A23(2,2)+
x      A23J2I(1,1)*A23(2,1)
RJR2(1,3) = A23J2I(1,3)*A23(3,3)+A23J2I(1,2)*A23(3,2)+
x      A23J2I(1,1)*A23(3,1)
RJR2(2,2) = A23(2,3)*A23J2I(2,3)+A23(2,2)*A23J2I(2,2)+
x      A23(2,1)*A23J2I(2,1)
RJR2(2,3) = A23J2I(2,3)*A23(3,3)+A23J2I(2,2)*A23(3,2)+
x      A23J2I(2,1)*A23(3,1)
RJR2(3,3) = A23(3,3)*A23J2I(3,3)+A23(3,2)*A23J2I(3,2)+
x      A23(3,1)*A23J2I(3,1)

```

C upper triangle elements of symmetric HTDIH

```

S11 = MU12+RJR2(1,1)+RJR1(1,1)
S12 = RJR2(1,2)+RJR1(1,2)
S13 = RJR2(1,3)+RJR1(1,3)
S22 = MU12+RJR2(2,2)+RJR1(2,2)
S23 = RJR2(2,3)+RJR1(2,3)
S33 = MU12+RJR2(3,3)+RJR1(3,3)

```

C upper triangle elements of symmetric HTDIH^-1

```

CF11 = S22*S33 - S23*S23
CF12 = S12*S33 - S13*S23
CF13 = S12*S23 - S13*S22
CF22 = S11*S33 - S13*S13
CF23 = S23*S11 - S12*S13
CF33 = S11*S22 - S12*S12
DET = S11*CF11 - S12*CF12 + S13*CF13
SI11 = CF11/DET
SI12 = -CF12/DET
SI13 = CF13/DET
SI22 = CF22/DET
SI23 = -CF23/DET

```



```

      SI33 = CF33/DET

C   cable force on helicopter, inertial axes components
      FC1N(1) = -HTDIFO(3)*SI13-HTDIFO(2)*SI12-HTDIFO(1)*SI11
      FC1N(2) = -HTDIFO(3)*SI23-HTDIFO(2)*SI22-HTDIFO(1)*SI12
      FC1N(3) = -HTDIFO(3)*SI33-HTDIFO(2)*SI23-HTDIFO(1)*SI13

      ELSE

C   elastic cables: fc12 = sum(tauj*kcj2, j = 1,...,nc)
      FC12(1) = 0.
      FC12(2) = 0.
      FC12(3) = 0.

      DO 300 J = 1,NC
      RAJ2(1,J) = RA2S2(1) + R2SJ2(1,J)
      RAJ2(2,J) = RA2S2(2) + R2SJ2(2,J)
      RAJ2(3,J) = RA2S2(3) + R2SJ2(3,J)
      LCJ(J) = SQRT(RAJ2(3,J)**2+RAJ2(2,J)**2+RAJ2(1,J)**2)
      KCJ2(1,J) = RAJ2(1,J)/LCJ(J)
      KCJ2(2,J) = RAJ2(2,J)/LCJ(J)
      KCJ2(3,J) = RAJ2(3,J)/LCJ(J)
      DLCJ(J) = DRA2S2(3)*KCJ2(3,J)+DRA2S2(2)*KCJ2(2,J)
X      +DRA2S2(1)*KCJ2(1,J)
      TAUJ(J) = AMAX1(0.,KS*(LCJ(J) - LCJO(J)) + CS*DLCJ(J))
      FC12(1) = FC12(1) + KCJ2(1,J)*TAUJ(J)
      FC12(2) = FC12(2) + KCJ2(2,J)*TAUJ(J)
      FC12(3) = FC12(3) + KCJ2(3,J)*TAUJ(J)
300    CONTINUE

      FC1N(1) = TN2(1,3)*FC12(3)+TN2(1,2)*FC12(2)+FC12(1)*TN2(1,1)
      FC1N(2) = TN2(2,3)*FC12(3)+FC12(2)*TN2(2,2)+FC12(1)*TN2(2,1)
      FC1N(3) = FC12(3)*TN2(3,3)+FC12(2)*TN2(3,2)+FC12(1)*TN2(3,1)

      ENDIF

C   in sim IC use susp force result from load-susp IC logic
      IF (IMODE.LT.0) THEN
      FC1N(1) = -TN2(1,3)*FC22(3)-TN2(1,2)*FC22(2)-FC22(1)*TN2(1,1)
      FC1N(2) = -TN2(2,3)*FC22(3)-FC22(2)*TN2(2,2)-FC22(1)*TN2(2,1)
      FC1N(3) = -FC22(3)*TN2(3,3)-FC22(2)*TN2(3,2)-FC22(1)*TN2(3,1)
      END IF

C   suspension forces and cg moments on the HC and load
      FC2N(1) = -FC1N(1)
      FC2N(2) = -FC1N(2)
      FC2N(3) = -FC1N(3)
      MC11(1) = FC1N(3)*A22(3,1)+FC1N(2)*A22(2,1)+FC1N(1)*A22(1,1)
      MC11(2) = FC1N(3)*A22(3,2)+FC1N(2)*A22(2,2)+FC1N(1)*A22(1,2)
      MC11(3) = FC1N(3)*A22(3,3)+FC1N(2)*A22(2,3)+FC1N(1)*A22(1,3)
      MC22(1) = FC1N(3)*A23(3,1)+FC1N(2)*A23(2,1)+FC1N(1)*A23(1,1)
      MC22(2) = FC1N(3)*A23(3,2)+FC1N(2)*A23(2,2)+FC1N(1)*A23(1,2)
      MC22(3) = FC1N(3)*A23(3,3)+FC1N(2)*A23(2,3)+FC1N(1)*A23(1,3)

```

C\*\*\* SEC 500: ACCELERATIONS.  $du = A^{-1} D^{-1} (fo + fc) = A^{-1} sf$

```

SF1N(1) = (FO1N(1) + FC1N(1))/M1
SF1N(2) = (FO1N(2) + FC1N(2))/M1
SF1N(3) = (FO1N(3) + FC1N(3))/M1
SF2N(1) = (FO2N(1) + FC2N(1))/M2
SF2N(2) = (FO2N(2) + FC2N(2))/M2
SF2N(3) = (FO2N(3) + FC2N(3))/M2
SUMM11(1) = MO11(1) + MC11(1)
SUMM11(2) = MO11(2) + MC11(2)
SUMM11(3) = MO11(3) + MC11(3)
SUMM22(1) = MO22(1) + MC22(1)
SUMM22(2) = MO22(2) + MC22(2)
SUMM22(3) = MO22(3) + MC22(3)
SM11(1) = SUMM11(3)*II1XZ + SUMM11(1)*II1XX
SM11(2) = SUMM11(2)*II1YY
SM11(3) = SUMM11(3)*II1ZZ + SUMM11(1)*II1XZ
SM22(1) = SUMM22(3)*II2XZ + SUMM22(1)*II2XX
SM22(2) = SUMM22(2)*II2YY
SM22(3) = SUMM22(3)*II2ZZ + SUMM22(1)*II2XZ
DV1SN(1) = SF1N(1)
DV1SN(2) = SF1N(2)
DV1SN(3) = SF1N(3)
DOM11(1) = SM11(1)
DOM11(2) = SM11(2)
DOM11(3) = SM11(3)
DOM22(1) = SM22(1)
DOM22(2) = SM22(2)
DOM22(3) = SM22(3)
TV5N(1) = A22(1,3)*SM11(3)+A22(1,2)*SM11(2)+SM11(1)*A22(1,1)
TV5N(2) = A22(2,3)*SM11(3)+SM11(2)*A22(2,2)+SM11(1)*A22(2,1)
TV5N(3) = SM11(3)*A22(3,3)+SM11(2)*A22(3,2)+SM11(1)*A22(3,1)
TV6N(1) = A23(1,3)*SM22(3)+A23(1,2)*SM22(2)+SM22(1)*A23(1,1)
TV6N(2) = A23(2,3)*SM22(3)+SM22(2)*A23(2,2)+SM22(1)*A23(2,1)
TV6N(3) = SM22(3)*A23(3,3)+SM22(2)*A23(3,2)+SM22(1)*A23(3,1)
DDRA2SN(1) = SF2N(1) - TV6N(1) - TV5N(1) - SF1N(1)
DDRA2SN(2) = SF2N(2) - TV6N(2) - TV5N(2) - SF1N(2)
DDRA2SN(3) = SF2N(3) - TV6N(3) - TV5N(3) - SF1N(3)
DDRA2S2(1) = T2N(1,3)*DDRA2SN(3)+T2N(1,2)*DDRA2SN(2)+
x DDRA2SN(1)*T2N(1,1)
DDRA2S2(2) = T2N(2,3)*DDRA2SN(3)+DDRA2SN(2)*T2N(2,2)+
x DDRA2SN(1)*T2N(2,1)
DDRA2S2(3) = DDRA2SN(3)*T2N(3,3)+DDRA2SN(2)*T2N(3,2)+
x DDRA2SN(1)*T2N(3,1)

```

C\*\*\* SEC 600. system cg errors, initialize integrators

```

R1S2SN(1) = TN2(1,3)*RA2S2(3)+TN1(1,3)*R1SA1(3)+
x TN2(1,2)*RA2S2(2)+TN1(1,2)*R1SA1(2)+
x RA2S2(1)*TN2(1,1)+R1SA1(1)*TN1(1,1)
R1S2SN(2) = TN2(2,3)*RA2S2(3)+TN1(2,3)*R1SA1(3)+
x RA2S2(2)*TN2(2,2)+R1SA1(2)*TN1(2,2)+
x RA2S2(1)*TN2(2,1)+R1SA1(1)*TN1(2,1)
R1S2SN(3) = RA2S2(3)*TN2(3,3)+R1SA1(3)*TN1(3,3)+

```

```

x          RA2S2(2)*TN2(3,2)+R1SA1(2)*TN1(3,2)+
x          RA2S2(1)*TN2(3,1)+R1SA1(1)*TN1(3,1)
RSN(1) = R1S2SN(1)*M2OM + R1SN(1)
RSN(2) = R1S2SN(2)*M2OM + R1SN(2)
RSN(3) = R1S2SN(3)*M2OM + R1SN(3)
VSN(1) = V1S2SN(1)*M2OM + V1SN(1)
VSN(2) = V1S2SN(2)*M2OM + V1SN(2)
VSN(3) = V1S2SN(3)*M2OM + V1SN(3)
TV7N(1) = A22(1,3)*DOM11(3)+A22(1,2)*DOM11(2)+DOM11(1)*A22(1,1)
TV7N(2) = A22(2,3)*DOM11(3)+DOM11(2)*A22(2,2)+DOM11(1)*A22(2,1)
TV7N(3) = DOM11(3)*A22(3,3)+DOM11(2)*A22(3,2)+DOM11(1)*A22(3,1)
TV8N(1) = A23(1,3)*DOM22(3)+A23(1,2)*DOM22(2)+DOM22(1)*A23(1,1)
TV8N(2) = A23(2,3)*DOM22(3)+DOM22(2)*A23(2,2)+DOM22(1)*A23(2,1)
TV8N(3) = DOM22(3)*A23(3,3)+DOM22(2)*A23(3,2)+DOM22(1)*A23(3,1)
ADU2N(1) = TV8N(1) + TV7N(1) + DDRA2SN(1)
ADU2N(2) = TV8N(2) + TV7N(2) + DDRA2SN(2)
ADU2N(3) = TV8N(3) + TV7N(3) + DDRA2SN(3)
DV1S2SN(1) = DAU2N(1) + ADU2N(1)
DV1S2SN(2) = DAU2N(2) + ADU2N(2)
DV1S2SN(3) = DAU2N(3) + ADU2N(3)
DVS(1) = DV1S2SN(1)*M2OM + DV1SN(1)
DVS(2) = DV1S2SN(2)*M2OM + DV1SN(2)
DVS(3) = DV1S2SN(3)*M2OM + DV1SN(3)
SSFN(1) = (FA2N(1) + FA1N(1))/M1PM2
SSFN(2) = (FA2N(2) + FA1N(2))/M1PM2
SSFN(3) = (FA2N(3) + FA1N(3))/M1PM2 + G

```

C save initial c.g. states and initialize integrations

```

      IF (T.LE.DTO2) THEN
        DO 600 I = 1,3
          RSON(I) = RSN(I)
          VSON(I) = VSN(I)
          ISSFN(I) = 0.
          IISSFN(I) = 0.
          OSSFN(I) = SSFN(I)
          OISSFN(I) = 0.
          DQ(I) = V1SN(I)
600      DQ(9+I) = DRA2S2(I)

          DPS1 = (OM11(2)*SPH1 + OM11(3)*CPH1)/CTH1
          DPH1 = OM11(1) + DPS1*STH1
          DTH1 = OM11(2)*CPH1 - OM11(3)*SPH1
          DPS2 = (OM22(2)*SPH2 + OM22(3)*CPH2)/CTH2
          DPH2 = OM22(1) + DPS2*STH2
          DTH2 = OM22(2)*CPH2 - OM22(3)*SPH2

          DO 601 I = 1,12
            ODU(I) = DU(I)
601      ODQ(I) = DQ(I)
        END IF

```

C cg errors and integrate SSFN for cg errors

```

      DELDVSN(1) = DVS(1) - SSFN(1)
      DELDVSN(2) = DVS(2) - SSFN(2)

```

```

DELDVSN(3) = DVS(3) - SSFN(3)
DELVS(1) = VS(1) - ISSFN(1) - VSON(1)
DELVS(2) = VS(2) - ISSFN(2) - VSON(2)
DELVS(3) = VS(3) - ISSFN(3) - VSON(3)
DELRN(1) = RS(1) - VSON(1)*T - RSON(1) - IISSFN(1)
DELRN(2) = RS(2) - VSON(2)*T - RSON(2) - IISSFN(2)
DELRN(3) = RS(3) - VSON(3)*T - RSON(3) - IISSFN(3)

```

```

DO 603 I = 1,3
ISSFN(I) = (3*SSFN(I) - OSSFN(I))*DTO2 + ISSFN(I)
IISSFN(I) = (OISSFN(I) + ISSFN(I))*DTO2 + IISSFN(I)
OSSFN(I) = SSFN(I)
603 OISSFN(I) = ISSFN(I)

```

C\*\*\* SEC 700. transfer data to calling program and data storage

```

FC11(1) = T1N(1,3)*FC1N(3)+T1N(1,2)*FC1N(2)+FC1N(1)*T1N(1,1)
FC11(2) = T1N(2,3)*FC1N(3)+FC1N(2)*T1N(2,2)+FC1N(1)*T1N(2,1)
FC11(3) = FC1N(3)*T1N(3,3)+FC1N(2)*T1N(3,2)+FC1N(1)*T1N(3,1)
SLINGTENSION = SQRT(FC1N(1)**2 + FC1N(2)**2 + FC1N(3)**2)

```

```

IF ((NP.EQ.NSTORE).AND.(IMODE.EQ.1)) THEN
FC22(1) = T2N(1,3)*FC2N(3)+T2N(1,2)*FC2N(2)+FC2N(1)*T2N(1,1)
FC22(2) = T2N(2,3)*FC2N(3)+FC2N(2)*T2N(2,2)+FC2N(1)*T2N(2,1)
FC22(3) = FC2N(3)*T2N(3,3)+FC2N(2)*T2N(3,2)+FC2N(1)*T2N(3,1)
V1S1(1) = T1N(1,3)*V1SN(3)+T1N(1,2)*V1SN(2)+V1SN(1)*T1N(1,1)
V1S1(2) = T1N(2,3)*V1SN(3)+V1SN(2)*T1N(2,2)+V1SN(1)*T1N(2,1)
V1S1(3) = V1SN(3)*T1N(3,3)+V1SN(2)*T1N(3,2)+V1SN(1)*T1N(3,1)

```

```

KCN(1) = FC1N(1)/SLINGTENSION
KCN(2) = FC1N(2)/SLINGTENSION
KCN(3) = FC1N(3)/SLINGTENSION
RA2SN(1) = TN2(1,3)*RA2S2(3)+TN2(1,2)*RA2S2(2)+
x RA2S2(1)*TN2(1,1)
RA2SN(2) = TN2(2,3)*RA2S2(3)+RA2S2(2)*TN2(2,2)+
x RA2S2(1)*TN2(2,1)
RA2SN(3) = RA2S2(3)*TN2(3,3)+RA2S2(2)*TN2(3,2)+
x RA2S2(1)*TN2(3,1)
ABSRA2S = SQRT(RA2S2(3)**2+RA2S2(2)**2+RA2S2(1)**2)
KCPN(1) = RA2SN(1)/ABSRA2S
KCPN(2) = RA2SN(2)/ABSRA2S
KCPN(3) = RA2SN(3)/ABSRA2S
KCXKCPN(1) = KCN(2)*KCPN(3) - KCPN(2)*KCN(3)
KCXKCPN(2) = KCPN(1)*KCN(3) - KCN(1)*KCPN(3)
KCXKCPN(3) = KCN(1)*KCPN(2) - KCPN(1)*KCN(2)
ANGKCKCP = ASIN(SQRT(KCXKCPN(1)**2+KCXKCPN(2)**2+
x KCXKCPN(3)**2))
KCXK2N(1) = KCN(2)*K2N(3) - K2N(2)*KCN(3)
KCXK2N(2) = K2N(1)*KCN(3) - KCN(1)*K2N(3)
KCXK2N(3) = KCN(1)*K2N(2) - K2N(1)*KCN(2)
ANGKCK2 = ASIN(SQRT(KCXK2N(1)**2+KCXK2N(2)**2+KCXK2N(3)**2))

```

C direction angles of FC11 ... first roll to the plane of the hook,  
C then pitch. This reverses usual euler sequence.

```

RPHCP      = -ATAN(FC11(2)/FC11(3))
RTHCP      =  ASIN(FC11(1)/SLINGTENSION)

C  direction angles of ra2s wrt HC heading axes
SPS1       = SIN(PS1)
CPS1       = COS(PS1)
KCPH(1)    = KCPN(2)*SPS1 + KCPN(1)*CPS1
KCPH(2)    = KCPN(2)*CPS1 - KCPN(1)*SPS1
PHCH       = -ASIN(KCPH(2))
THCH       =  ASIN(KCPH(1)/COS(PHCH))
RA2S1(1)   = T1N(1,3)*RA2SN(3)+T1N(1,2)*RA2SN(2)+
x           RA2SN(1)*T1N(1,1)
RA2S1(2)   = T1N(2,3)*RA2SN(3)+RA2SN(2)*T1N(2,2)+
x           RA2SN(1)*T1N(2,1)
RA2S1(3)   = RA2SN(3)*T1N(3,3)+RA2SN(2)*T1N(3,2)+
x           RA2SN(1)*T1N(3,1)
KCP1(1)    = RA2S1(1)/ABSRA2S
KCP1(2)    = RA2S1(2)/ABSRA2S
KCP1(3)    = RA2S1(3)/ABSRA2S

C  direction angles of ra2s wrt HC body axes
RPHC       = -ASIN(KCP1(2))
RTHC       =  ASIN(KCP1(1)/COS(RPHC))

C  inertial direction angles and rates for ra2s.
PHC        = -ASIN(KCPN(2))
CPHC       = COS(PHC)
SPHC       = SIN(PHC)
THC        = ASIN(KCPN(1)/CPHC)
CTHC       = COS(THC)
STHC       = SIN(THC)
DNRA2S2(1) = OM22(2)*RA2S2(3)-RA2S2(2)*OM22(3)+DRA2S2(1)
DNRA2S2(2) = -OM22(1)*RA2S2(3)+RA2S2(1)*OM22(3)+DRA2S2(2)
DNRA2S2(3) = DRA2S2(3)+OM22(1)*RA2S2(2)-RA2S2(1)*OM22(2)
DNRA2SN(1) = TN2(1,3)*DNRA2S2(3)+TN2(1,2)*DNRA2S2(2)+
x           DNRA2S2(1)*TN2(1,1)
DNRA2SN(2) = TN2(2,3)*DNRA2S2(3)+DNRA2S2(2)*TN2(2,2)+
x           DNRA2S2(1)*TN2(2,1)
DNRA2SN(3) = DNRA2S2(3)*TN2(3,3)+DNRA2S2(2)*TN2(3,2)+
x           DNRA2S2(1)*TN2(3,1)
DRA2S      = KCPN(1)*DNRA2SN(1)+KCPN(2)*DNRA2SN(2)+
x           KCPN(3)*DNRA2SN(3)
DPHC       = -(DNRA2SN(2)-DRA2S*KCPN(2))/(ABSRA2S*CPHC)
DTHC       = (DNRA2SN(1)-DRA2S*KCPN(1)+ABSRA2S*SPHC*STHC*DPHC)/
x           (ABSRA2S*CPHC*CTHC)

C  load position
R2SN(1)    = R1SN(1) + R1S2SN(1)
R2SN(2)    = R1SN(2) + R1S2SN(2)
R2SN(3)    = R1SN(3) + R1S2SN(3)

C  load acceleration, body axes components
DV2SN(1)   = DV1SN(1) + DV1S2SN(1)
DV2SN(2)   = DV1SN(2) + DV1S2SN(2)

```

```

DV2SN(3) = DV1SN(3) + DV1S2SN(3)
DV2S2(1) = T2N(1,3)*DV2SN(3)+T2N(1,2)*DV2SN(2)+
x      DV2SN(1)*T2N(1,1)
DV2S2(2) = T2N(2,3)*DV2SN(3)+DV2SN(2)*T2N(2,2)+
x      DV2SN(1)*T2N(2,1)
DV2S2(3) = DV2SN(3)*T2N(3,3)+DV2SN(2)*T2N(3,2)+
x      DV2SN(1)*T2N(3,1)

```

C parts of dv2sn: hook accln, centrifugal, stretch, and dom22 parts.

```

DVAN(1) = TN1(1,3)*CA11(3)+TN1(1,2)*CA11(2)+CA11(1)*TN1(1,1)
x      + TV7N(1) + DV1SN(1)
DVAN(2) = TN1(2,3)*CA11(3)+CA11(2)*TN1(2,2)+CA11(1)*TN1(2,1)
x      + TV7N(2) + DV1SN(2)
DVAN(3) = CA11(3)*TN1(3,3)+CA11(2)*TN1(3,2)+CA11(1)*TN1(3,1)
x      + TV7N(3) + DV1SN(3)
OMXR22(1) = OM22(2)*RA2S2(3) - RA2S2(2)*OM22(3)
OMXR22(2) = -OM22(1)*RA2S2(3) + RA2S2(1)*OM22(3)
OMXR22(3) = OM22(1)*RA2S2(2) - RA2S2(1)*OM22(2)
DV2SP2(1) = OM22(2)*OMXR22(3) - OMR22(2)*OM22(3)
DV2SP2(2) = OMR22(1)*OM22(3) - OM22(1)*OMXR22(3)
DV2SP2(3) = OM22(1)*OMXR22(2) - OMR22(1)*OM22(2)
DV2SPN(1) = TN2(1,3)*DV2SP2(3)+TN2(1,2)*DV2SP2(2)+
x      DV2SP2(1)*TN2(1,1)
DV2SPN(2) = TN2(2,3)*DV2SP2(3)+DV2SP2(2)*TN2(2,2)+
x      DV2SP2(1)*TN2(2,1)
DV2SPN(3) = DV2SP2(3)*TN2(3,3)+DV2SP2(2)*TN2(3,2)+
x      DV2SP2(1)*TN2(3,1)
OMXDR22(1) = OM22(2)*DRA2S2(3) - DRA2S2(2)*OM22(3)
OMXDR22(2) = DRA2S2(1)*OM22(3) - OM22(1)*DRA2S2(3)
OMXDR22(3) = OM22(1)*DRA2S2(2) - DRA2S2(1)*OM22(2)
OMXDR2N(1) = TN2(1,3)*OMXDR22(3)+TN2(1,2)*OMXDR22(2)+
x      OMXDR22(1)*TN2(1,1)
OMXDR2N(2) = TN2(2,3)*OMXDR22(3)+OMXDR22(2)*TN2(2,2)+
x      OMXDR22(1)*TN2(2,1)
OMXDR2N(3) = OMXDR22(3)*TN2(3,3)+OMXDR22(2)*TN2(3,2)+
x      OMXDR22(1)*TN2(3,1)
DV2SSN(1) = DDRA2SN(1) + 2.*OMXDR2N(1)
DV2SSN(2) = DDRA2SN(2) + 2.*OMXDR2N(2)
DV2SSN(3) = DDRA2SN(3) + 2.*OMXDR2N(3)
DV2SCHKN(1) = DVAN(1) + DV2SPN(1) + TV8N(1) + DV2SSN(1)
DV2SCHKN(2) = DVAN(2) + DV2SPN(2) + TV8N(2) + DV2SSN(2)
DV2SCHKN(3) = DVAN(3) + DV2SPN(3) + TV8N(3) + DV2SSN(3)
DV2SPON(1) = DV2SPN(1) + TV8N(1)
DV2SPON(2) = DV2SPN(2) + TV8N(2)
DV2SPON(3) = DV2SPN(3) + TV8N(3)

```

C load ang vel in HC body axes

```

OM2N(1) = TN2(1,3)*OM22(3)+TN2(1,2)*OM22(2)+OM22(1)*TN2(1,1)
OM2N(2) = TN2(2,3)*OM22(3)+OM22(2)*TN2(2,2)+OM22(1)*TN2(2,1)
OM2N(3) = OM22(3)*TN2(3,3)+OM22(2)*TN2(3,2)+OM22(1)*TN2(3,1)
OM21(1) = T1N(1,3)*OM2N(3)+T1N(1,2)*OM2N(2)+OM2N(1)*T1N(1,1)
OM21(2) = T1N(2,3)*OM2N(3)+OM2N(2)*T1N(2,2)+OM2N(1)*T1N(2,1)
OM21(3) = OM2N(3)*T1N(3,3)+OM2N(2)*T1N(3,2)+OM2N(1)*T1N(3,1)

```

C load apparent gravity, apparent magnetic dip angle and

```

C fluxgate compass correction
  GMA2SN(1)= - DV2SN(1)
  GMA2SN(2)= - DV2SN(2)
  GMA2SN(3)= G - DV2SN(3)
  GMA2S2(1)= G*KN2(1) - DV2S2(1)
  GMA2S2(2)= G*KN2(2) - DV2S2(2)
  GMA2S2(3)= G*KN2(3) - DV2S2(3)
  ABSGMA2S = SQRT(GMA2SN(3)**2+GMA2SN(2)**2+GMA2SN(1)**2)
  PHA2N    = -ASIN(GMA2SN(2)/ABSGMA2S)
  THA2N    = ATAN(GMA2SN(1)/GMA2SN(3))
  PHA22    = -ASIN(GMA2S2(2)/ABSGMA2S)
  THA22    = ATAN(GMA2S2(1)/GMA2S2(3))
  ANGKAKN  = ASIN(SQRT(GMA2SN(2)**2 + GMA2SN(1)**2)/ABSGMA2S)
  ANGKAK2  = ASIN(SQRT(GMA2S2(2)**2 + GMA2S2(1)**2)/ABSGMA2S)
  CPHA2N   = COS(PHA2N)
  SPHA2N   = SIN(PHA2N)
  CTHA2N   = COS(THA2N)
  STHA2N   = SIN(THA2N)
  DELPSM   = ATAN2(SDIP*SPHA2N,CDIP-CPHA2N*STHA2N*SIN(DIP+THA2N))
  SDIPP    = (CDIP*GMA2SN(1) + SDIP*GMA2SN(3))/ABSGMA2S
  DIPP     = ASIN(SDIPP)

C load ang vel magnitudes and airspeed
  ABSOM2   = SQRT(OM22(3)**2 + OM22(2)**2 + OM22(1)**2)
  ABSPQ2   = SQRT(OM22(1)**2 + OM22(2)**2)
  ABSDV2S  = SQRT(DV2SN(3)**2 + DV2SN(2)**2 + DV2SN(1)**2)
  ABSVA2S  = SQRT(VA2SN(3)**2 + VA2SN(2)**2 + VA2SN(1)**2)

C load p, q in load-HC heading axes per MT
  CPS2M1   = COS(PS2 - PS1)
  SPS2M1   = SIN(PS2 - PS1)
  P2P      = OM22(1)*CPS2M1 - OM22(2)*SPS2M1
  Q2P      = OM22(1)*SPS2M1 + OM22(2)*CPS2M1
  NS = NS + 1
  NP = 0
ENDIF

C*** SEC 800. INTEGRATION: compute u, q at tn + dt, store past values

C update u. null stretching rate exactly if inelastic suspension
  DO 800 I = 1,12
    800 U(I) = U(I) + (3*DU(I) - ODU(I))*DT02
    IF (STRETCH.EQ.0) THEN
      DRA2S2(1) = 0.
      DRA2S2(2) = 0.
      DRA2S2(3) = 0.
    END IF
  C compute dq(q,u) using updated u
  DO 801 I = 1,3
    DQ(I) = V1SN(I)
  801 DQ(I+9) = DRA2S2(I)

  DPS1 = (OM11(2)*SPH1 + OM11(3)*CPH1)/CTH1
  DPH1 = OM11(1) + DPS1*STH1

```

```

DTH1  = OM11(2)*CPH1 - OM11(3)*SPH1
DPS2  = (OM22(2)*SPH2 + OM22(3)*CPH2)/CTH2
DPH2  = OM22(1) + DPS2*STH2
DTH2  = OM22(2)*CPH2 - OM22(3)*SPH2

C  update q and store past values
DO 802 I = 1,12
  Q(I)  = Q(I) + (ODQ(I) + DQ(I))*DT02
  ODU(I) = DU(I)
802 ODQ(I) = DQ(I)

  IF (IMODE.EQ.1) NP = NP + 1

END

```



## B.7. SUBROUTINE ghslmc\_ic

**Functions:** Initializes the load suspension for the single lift, multi-cable suspension configuration. Trim attitude is computed, with or without cable stretching.

**Inputs:** RA2P02, R2P2SO2, R2P2S2, other slung load variables in slvars.cmn

**Outputs:** RA2S2, PH2, TH2, PS2

**Called By:** ghslmc

**Calls:** conexaero, wake

**Comments:** Originally written as slmc\_ic for SL\_DRIVER. Iterative solution for trim attitude and cable positions, with error messages generated if no convergence.

### Program Listing:

```
C  subroutine ghslmc_ic.f

C  load-suspension initialization for multicable suspension.
C  LSCicolani, 12 sept 96, modified for GENHEL, 06 aug 98, PTyson
C  INPUTS are
C    Ra2'o2 = Ra2'2 for inelastic sling or loaded elastic sling at
C             hover and nominal cg location.
C    R2'2*o2 = cg offset fm R2' for inelastic sling or nominal offset
C             for elastic sling corresponding to equal cable loading
C             in hover.
C    R2'2*2 = cg offset from R2' for elastic sling for simn run.
C  For inelastic sling solve: GIVEN ra2'o2, r2'2*o2, va2*n FIND trim
C  attitude.
C  For elastic sling solve for trim attitude and ra2*2 in 2 steps;
C    (1) GIVEN nominal hover values of ra2'o2, r2'2*o2 FIND {lcjo}
C    (2) GIVEN {lcjo}, va2*n, r2'2*2 FIND trim attitude and Ra2*2.
C  Step 2 accounts for conditions (Va*n, cg offset) different from the
C  the design conditions of the rigging procedure (hover, nominal cg).
      SUBROUTINE GHSLMC_IC

      INCLUDE 'slvars.cmn'
      REAL RA2PP2(3), RA2PPP2(3), NCRA2P2(3), V1SN(3), RAJ2(3,8),
X        KCJ2(3,8), KJ2(3,3), ERRA1, ERRA3, ERRR2, DOV, NQSL, MAXG,
X        SUMSM2(3), SUMFC22(3), FCR2(3), FCP2(3), FCP12(3), RHS2(3)
      INTEGER JP(3), JR(5)

      EQUIVALENCE (A( 6), PS1 )
      EQUIVALENCE (A(64), V1SN(1) )
C *****
C  reference air velocity vector and inertial direction angles
      VA2SN(1) = V1SN(1) - WN(1)
      VA2SN(2) = V1SN(2) - WN(2)
      VA2SN(3) = V1SN(3) - WN(3)
      VA2XY2   = VA2SN(1)**2 + VA2SN(2)**2
      VA22     = VA2XY2 + VA2SN(3)**2
      VA2XY    = SQRT(VA2XY2)
```

```

VA2      = SQRT(VA22)
QSL      = .5*RHO*VA22
GAM      = 0.
IF (VA2.GT.2) GAM = ASIN(-VA2SN(3)/VA2)
SGAM     = SIN(GAM)
CGAM     = COS(GAM)
PSVA     = PS1
IF (VA2XY.GT.2.) PSVA = ATAN2(VA2SN(2),VA2SN(1))

```

C initial hook-to-load-cg coords: inelastic/elastic sling at nom hover

```

RA2S2(1) = RA2PO2(1) + R2P2SO2(1)
RA2S2(2) = RA2PO2(2) + R2P2SO2(2)
RA2S2(3) = RA2PO2(3) + R2P2SO2(3)

```

C\*\*\* SEC 100. INELASTIC SLING TRIM \*\*\*\*\*  
C GIVEN Ra2\*2, VA2\*n, FIND load attitude from load moment balance eqn.  
C ps2 is selected to place load load x-axis 90deg from ref direction  
C (ps1 if hover, else psva). This places the load broadside to the  
C wind or takeoff direction (and same all elongated loads for which  
C x-axis in aero model is aligned w/ elongated direction). This should  
C also be the trim load yaw angle for loads in hover, for the MILVAN at  
C all airspeeds, and for loads w no aero moments.

```

IF (STRETCH.EQ.0) THEN
ITR81 = 0
PH2   = 0.
TH2   = 0.
PS2   = PSVA
CPH2  = 1.
SPH2  = 0.
CTH2  = 1.
STH2  = 0.

```

C Begin iteration

```

100 CONTINUE
ITR81 = ITR81 + 1
PH2P   = PH2
TH2P   = TH2
PS2P   = PS2
CPS2   = COS(PS2)
SPS2   = SIN(PS2)
T2N(1,1) = CPS2*CTH2
T2N(1,2) = CTH2*SPS2
T2N(1,3) = -STH2
T2N(2,1) = CPS2*SPH2*STH2-CPH2*SPS2
T2N(2,2) = SPH2*SPS2*STH2+CPH2*CPS2
T2N(2,3) = CTH2*SPH2
T2N(3,1) = CPH2*CPS2*STH2+SPH2*SPS2
T2N(3,2) = CPH2*SPS2*STH2-CPS2*SPH2
T2N(3,3) = CPH2*CTH2
VA2SN(1) = V1SN(1) - WN(1)
VA2SN(2) = V1SN(2) - WN(2)
VA2SN(3) = V1SN(3) - WN(3)
VA2S2(1) = T2N(1,3)*VA2SN(3)+T2N(1,2)*VA2SN(2)+VA2SN(1)*T2N(1,1)

```

```

      VA2S2(2) = T2N(2,3)*VA2SN(3)+VA2SN(2)*T2N(2,2)+VA2SN(1)*T2N(2,1)
      VA2S2(3) = VA2SN(3)*T2N(3,3)+VA2SN(2)*T2N(3,2)+VA2SN(1)*T2N(3,1)
C Load aero: option IAERSL = 0: no aero, 1: drag only, 2: CONEX aero
      DO 110 I = 1,3
      FA22(I) = 0.
110  MA22(I) = 0.
      IF (IWAKE.EQ.1) CALL WAKE
      IF (IAERSL.EQ.1) THEN
        VA2 = SQRT(VA2S2(1)**2 + VA2S2(2)**2 + VA2S2(3)**2)
        DOV = DOQ*RHO*VA2/2.
        FA22(1) = -DOV*VA2S2(1)
        FA22(2) = -DOV*VA2S2(2)
        FA22(3) = -DOV*VA2S2(3)
      ELSE IF (IAERSL.EQ.2) THEN
        CALL CONEXAERO
      END IF

C Compute attitude angles. Compute ps2 for beta = 0
      RHS2(1) = (-FA22(2)*RA2S2(3) + RA2S2(2)*FA22(3) + MA22(1))/W2
      RHS2(2) = ( FA22(1)*RA2S2(3) - RA2S2(1)*FA22(3) + MA22(2))/W2
      RHS2(3) = ( MA22(3) - FA22(1)*RA2S2(2) + RA2S2(1)*FA22(2))/W2
      SPH2 = (RHS2(1) + RA2S2(2)*CPH2*CTH2)/(RA2S2(3)*CTH2)
      STH2 = (RHS2(2) - RA2S2(1)*CPH2*CTH2)/RA2S2(3)
      PH2 = ASIN(SPH2)
      TH2 = ASIN(STH2)
      CPH2 = COS(PH2)
      CTH2 = COS(TH2)
      IF (CGAM.LT..01) THEN
        PS2 = PSVA
      ELSE
        SQSDPS = SQRT(CPH2**2 + (SPH2*STH2)**2)
        SDPSO = SPH2*STH2/SQSDPS
        SDPS1 = AMAX1(-1., AMIN1(SPH2*CTH2*SGAM/(CGAM*SQSDPS), 1.))
        DPSO = ASIN(SDPSO)
        DPS1 = ASIN(SDPS1)
        PS2 = PSVA + DPSO - DPS1
      END IF

      ERR1 = ABS(PH2-PH2P) + ABS(TH2-TH2P) + ABS(PS2-PS2P)
      IF ((ERR1.GT..0001).AND.(ITR81.LT.101)) GO TO 100
C residual load angular acceln
      FC22(1) = STH2*W2 - FA22(1)
      FC22(2) = -CTH2*SPH2*W2 - FA22(2)
      FC22(3) = -CTH2*CPH2*W2 - FA22(3)
      SUMSM2(1) = ( FC22(2)*RA2S2(3)-RA2S2(2)*FC22(3)+MA22(1))/I2XX
      SUMSM2(2) = (-FC22(1)*RA2S2(3)+RA2S2(1)*FC22(3)+MA22(2))/I2YY
      SUMSM2(3) = ( MA22(3)+FC22(1)*RA2S2(2)-RA2S2(1)*FC22(2))/I2ZZ

      GO TO 400
    ENDIF

C*** END OF INELASTIC SLING TRIM

```

```

C*****
C SEC 200. ELASTIC SLING. COMPUTE UNLOADED SLING LEG LENGTHS FROM
C NOMINAL HOVER GEOMETRY, RA2'O2, R2'2*O2 in 2 steps.

C FIRST, compute load attitude in hover via iterative solution
    ITR81 = 0
    PH2 = 0
    TH2 = 0
200  CONTINUE
    ITR81 = ITR81 + 1
    PH2P = PH2
    TH2P = TH2
    SPH2 = RA2S2(2)*CPH2/RA2S2(3)
    STH2 = -RA2S2(1)*CPH2*CTH2/RA2S2(3)
    PH2 = ASIN(SPH2)
    TH2 = ASIN(STH2)
    CPH2 = COS(PH2)
    CTH2 = COS(TH2)
    ERRA1 = ABS(PH2-PH2P) + ABS(TH2-TH2P)
    IF ((ERRA1.GT..0001).AND.(ITR81.LT.101)) GO TO 200
    FC22(1) = STH2*W2
    FC22(2) = -CTH2*SPH2*W2
    FC22(3) = -CTH2*CPH2*W2

C SECOND, solve for {lcjo}. Select tension in redundant cables, if any,
C assuming all cables carry equal force along load vertical, k2, for
C hover and nominal cg location; and solve for remaining cable tensions
C from cable force identity, FC22 = -sum(tauj*kcj).
    DO 260 J = 1,NC
    RAJ2(1,J) = RA2PO2(1) + R2PJ2(1,J)
    RAJ2(2,J) = RA2PO2(2) + R2PJ2(2,J)
    RAJ2(3,J) = RA2PO2(3) + R2PJ2(3,J)
    LCJ(J) = SQRT(RAJ2(3,J)**2 + RAJ2(2,J)**2 + RAJ2(1,J)**2)
    KCJ2(1,J) = RAJ2(1,J)/LCJ(J)
    KCJ2(2,J) = RAJ2(2,J)/LCJ(J)
    KCJ2(3,J) = RAJ2(3,J)/LCJ(J)
260  CONTINUE

C Initialize some elements
    JP(1) = 1
    JP(2) = 2
    JP(3) = 3
    FCR2(1) = 0.
    FCR2(2) = 0.
    FCR2(3) = 0.
    MAXG = 0.

    IF (NC.EQ.3) GO TO 280
C If nc > 3 there are redundant cables.
C Find the "principal" cables = 3 cables w max Gramian
    DO 210 I = 1, NC-2
    DO 210 J = I+1,NC-1
    DO 210 K = J+1,NC
    DO 220 I1 = 1,3

```

```

      KJ2(I1,1) = KCJ2(I1,I)
      KJ2(I1,2) = KCJ2(I1,J)
220  KJ2(I1,3) = KCJ2(I1,K)

C Elements of G = transpose(KJ2).KJ2 and its determinant (Gramian)
      G11 = 1.
      G12 = KJ2(3,1)*KJ2(3,2)+KJ2(2,1)*KJ2(2,2)+KJ2(1,1)*KJ2(1,2)
      G13 = KJ2(3,1)*KJ2(3,3)+KJ2(2,1)*KJ2(2,3)+KJ2(1,1)*KJ2(1,3)
      G22 = 1.
      G23 = KJ2(3,2)*KJ2(3,3)+KJ2(2,2)*KJ2(2,3)+KJ2(1,2)*KJ2(1,3)
      G33 = 1.
      CF11 = G22*G33 - G23*G23
      CF12 = G12*G33 - G13*G23
      CF13 = G12*G23 - G13*G22
      DETG = G11*CF11 - G12*CF12 + G13*CF13
      IF (DETG.GT.MAXG) THEN
        JP(1) = I
        JP(2) = J
        JP(3) = K
        MAXG = DETG
      END IF
210  CONTINUE

C Enumerate the nc - 3 redundant cables and assign their tensions to
C correspond to equal cable force component along k2 for all nc cables.
C FCR2 is the force on load carried by the redundant cables
      I = 1
      DO 230 J = 1,NC
        IF ((J.NE.JP(1)).AND.(J.NE.JP(2)).AND.(J.NE.JP(3))) THEN
          JR(I) = J
          I=I+1
        END IF
230  CONTINUE

      DO 240 I = 1,NC-3
        J = JR(I)
        TAUJ(J) = -FC22(3)/(KCJ2(3,J)*FLOAT(NC))
        FCR2(1) = -KCJ2(1,J)*TAUJ(J) + FCR2(1)
        FCR2(2) = -KCJ2(2,J)*TAUJ(J) + FCR2(2)
        FCR2(3) = -KCJ2(3,J)*TAUJ(J) + FCR2(3)
240  CONTINUE

C Compute stretch of principal cables from the cable force identity
280  DO 250 I = 1,3
      KJ2(I,1) = KCJ2(I,JP(1))
      KJ2(I,2) = KCJ2(I,JP(2))
      KJ2(I,3) = KCJ2(I,JP(3))
250  CONTINUE

C Elements of symmetric G = transpose(KJ2).KJ2
      G11 = 1.
      G12 = KJ2(3,1)*KJ2(3,2)+KJ2(2,1)*KJ2(2,2)+KJ2(1,1)*KJ2(1,2)
      G13 = KJ2(3,1)*KJ2(3,3)+KJ2(2,1)*KJ2(2,3)+KJ2(1,1)*KJ2(1,3)
      G22 = 1.

```

```

G23 = KJ2(3,2)*KJ2(3,3)+KJ2(2,2)*KJ2(2,3)+KJ2(1,2)*KJ2(1,3)
G33 = 1.

```

C Elements of symmetric inverse G

```

CF11 = G22*G33 - G23*G23
CF12 = G12*G33 - G13*G23
CF13 = G12*G23 - G13*G22
CF22 = G11*G33 - G13*G13
CF23 = G23*G11 - G12*G13
CF33 = G11*G22 - G12*G12
DETG = G11*CF11 - G12*CF12 + G13*CF13
GI11 = CF11/DETG
GI12 = -CF12/DETG
GI13 = CF13/DETG
GI22 = CF22/DETG
GI23 = -CF23/DETG
GI33 = CF33/DETG

```

C fcp = force on load due to "principal" cables

```

FCP2(1) = FC22(1) - FCR2(1)
FCP2(2) = FC22(2) - FCR2(2)
FCP2(3) = FC22(3) - FCR2(3)
FCP12(1) = FCP2(3)*KJ2(3,1)+FCP2(2)*KJ2(2,1)+FCP2(1)*KJ2(1,1)
FCP12(2) = FCP2(3)*KJ2(3,2)+FCP2(2)*KJ2(2,2)+FCP2(1)*KJ2(1,2)
FCP12(3) = FCP2(3)*KJ2(3,3)+FCP2(2)*KJ2(2,3)+FCP2(1)*KJ2(1,3)

```

C Principal cables: tensions, stretch, unloaded lengths

```

TAUJ(JP(1)) = -FCP12(3)*GI13-FCP12(2)*GI12-FCP12(1)*GI11
TAUJ(JP(2)) = -FCP12(3)*GI23-FCP12(2)*GI22-FCP12(1)*GI12
TAUJ(JP(3)) = -FCP12(3)*GI33-FCP12(2)*GI23-FCP12(1)*GI13
IF ((TAUJ(JP(1)).LT.0.).OR.(TAUJ(JP(2)).LT.0.).OR.
X (TAUJ(JP(3)).LT.0.)) TYPE *,
X 'STRETCH TRIM FOR NOMINAL HOVER FAILS. CHECK INPUT PARAMETERS'

```

C Unloaded cable lengths

```

DO 270 J = 1,NC
270 LCJO(J) = LCJ(J) - TAUJ(J)/KS

```

```

C*****
C SEC 300. ELASTIC SLING TRIM: TRIM FOR INITIAL CONDITIONS OF THIS RUN
C GIVEN {lcjo} FIND load attitude and Ra2*2. Double iteration: first
C find attitude given Ra2*2 fm load mom balance, then find Ra2*2 given
C attitude from cable force identity. Second, iterate these two steps.

```

C Begin outer loop iteration

```

ITR84 = 0
PS2 = PSVA
RA2P2(1) = RA2PO2(1)
RA2P2(2) = RA2PO2(2)
RA2P2(3) = RA2PO2(3)
310 CONTINUE

```

C Save past values for outer loop

```

ITR84 = ITR84 + 1

```

```

    RA2PPP2(1) = RA2P2(1)
    RA2PPP2(2) = RA2P2(2)
    RA2PPP2(3) = RA2P2(3)
    PH2PP = PH2
    TH2PP = TH2
    PS2PP = PS2
C Begin inner loop iteration for attitude given Ra2*2.
    ITR82 = 0
    CPH2 = COS(PH2)
    SPH2 = SIN(PH2)
    CTH2 = COS(TH2)
    STH2 = SIN(TH2)

C pht switched order to bring fol 3 lines ahead of continue statement
    RA2S2(1) = RA2P2(1) + R2P2S2(1)
    RA2S2(2) = RA2P2(2) + R2P2S2(2)
    RA2S2(3) = RA2P2(3) + R2P2S2(3)

320 CONTINUE
    ITR82 = ITR82 + 1
C Save past values for attitude iteration
    PH2P = PH2
    TH2P = TH2
    PS2P = PS2

C Load aerodynamics
    CPS2 = COS(PS2)
    SPS2 = SIN(PS2)
    T2N(1,1) = CPS2*CTH2
    T2N(1,2) = CTH2*SPS2
    T2N(1,3) = -STH2
    T2N(2,1) = CPS2*SPH2*STH2-CPH2*SPS2
    T2N(2,2) = SPH2*SPS2*STH2+CPH2*CPS2
    T2N(2,3) = CTH2*SPH2
    T2N(3,1) = CPH2*CPS2*STH2+SPH2*SPS2
    T2N(3,2) = CPH2*SPS2*STH2-CPS2*SPH2
    T2N(3,3) = CPH2*CTH2
    VA2SN(1) = V1SN(1) - WN(1)
    VA2SN(2) = V1SN(2) - WN(2)
    VA2SN(3) = V1SN(3) - WN(3)
    VA2S2(1) = T2N(1,3)*VA2SN(3)+T2N(1,2)*VA2SN(2)+VA2SN(1)*T2N(1,1)
    VA2S2(2) = T2N(2,3)*VA2SN(3)+VA2SN(2)*T2N(2,2)+VA2SN(1)*T2N(2,1)
    VA2S2(3) = VA2SN(3)*T2N(3,3)+VA2SN(2)*T2N(3,2)+VA2SN(1)*T2N(3,1)

C Load aero: option IAERSL = 0: no aero, 1: drag only, 2: CONEX aero
    DO 340 I = 1,3
    FA22(I) = 0.
340 MA22(I) = 0.
    IF (IWAKE.EQ.1) CALL WAKE
    IF (IAERSL.EQ.1) THEN
        VA2 = SQRT(VA2S2(1)**2 + VA2S2(2)**2 + VA2S2(3)**2)
        DOV = DOQ*RHO*VA2/2.
        FA22(1) = -DOV*VA2S2(1)
        FA22(2) = -DOV*VA2S2(2)

```

```

        FA22(3) = -DOV*VA2S2(3)
ELSE IF (IAERSL.EQ.2) THEN
        CALL CONEXAERO
END IF

C Compute pitch and roll from moment balance.
RHS2(1) = (-FA22(2)*RA2S2(3) + RA2S2(2)*FA22(3) + MA22(1))/W2
RHS2(2) = ( FA22(1)*RA2S2(3) - RA2S2(1)*FA22(3) + MA22(2))/W2
RHS2(3) = ( MA22(3) - FA22(1)*RA2S2(2) + RA2S2(1)*FA22(2))/W2
SPH2    = (RHS2(1) + RA2S2(2)*CPH2*CTH2)/(RA2S2(3)*CTH2)
STH2    = (RHS2(2) - RA2S2(1)*CPH2*CTH2)/RA2S2(3)
PH2     = ASIN(SPH2)
TH2     = ASIN(STH2)
CPH2    = COS(PH2)
CTH2    = COS(TH2)

C Compute ps2 such that beta = 0
IF (CGAM.LT.0.01) THEN
        PS2 = PSVA
ELSE
        SQSDPS = SQRT(CPH2**2 + (SPH2*STH2)**2)
        SDPSO  = SPH2*STH2/SQSDPS
        SDPS1  = AMAX1(-1., AMIN1(SPH2*CTH2*SGAM/(CGAM*SQSDPS), 1.))
        DPSO   = ASIN(SDPSO)
        DPS1   = ASIN(SDPS1)
        PS2    = PSVA + DPSO - DPS1
END IF

C Convergence test for attitude iteration
ERRA2   = ABS(PH2-PH2P) + ABS(TH2-TH2P) + ABS(PS2-PS2P)
IF ((ERRA2.GT..0001).AND.(ITR82.LT.101)) GO TO 320

C Begin inner loop iteration for Ra2'2 given FC22
ITR83   = 0
FC22(1) = STH2*W2 - FA22(1)
FC22(2) = -CTH2*SPH2*W2 - FA22(2)
FC22(3) = -CTH2*CPH2*W2 - FA22(3)
330 CONTINUE
ITR83 = ITR83+1

C Save past values for Ra2p2 iteration
RA2PP2(1) = RA2P2(1)
RA2PP2(2) = RA2P2(2)
RA2PP2(3) = RA2P2(3)

C Calculate cable directions for current Ra2p2
DO 350 J = 1,NC
        RAJ2(1,J) = RA2P2(1) + R2PJ2(1,J)
        RAJ2(2,J) = RA2P2(2) + R2PJ2(2,J)
        RAJ2(3,J) = RA2P2(3) + R2PJ2(3,J)
        LCJ(J)    = SQRT(RAJ2(3,J)**2 + RAJ2(2,J)**2 + RAJ2(1,J)**2)
        KCJ2(1,J) = RAJ2(1,J)/LCJ(J)
        KCJ2(2,J) = RAJ2(2,J)/LCJ(J)
        KCJ2(3,J) = RAJ2(3,J)/LCJ(J)
350 CONTINUE

```



```

C Compute Ra2p2 from cable force identity
  DO 360 I = 1,3
    NCRA2P2(I) = -FC22(I)/KS
  DO 361 J = 1,NC
361  NCRA2P2(I) = NCRA2P2(I) - R2PJ2(I,J) + KCJ2(I,J)*LCJO(J)
360  RA2P2(I) = NCRA2P2(I)/FLOAT(NC)

C Convergence test for stretch iteration
  ERRR1 = ABS(RA2P2(1)-RA2PP2(1)) + ABS(RA2P2(2)-RA2PP2(2))
X      + ABS(RA2P2(3)-RA2PP2(3))
  IF ((ERRR1.GT..0001).AND.(ITR83.LT.101)) GO TO 330

C Check outer loop for simultaneous convergence of attitude and stretch
  ERRA3 = ABS(PH2 - PH2PP) + ABS(TH2 - TH2PP) + ABS(PS2 - PS2PP)
  ERRR2 = ABS(RA2P2(1) - RA2PPP2(1)) + ABS(RA2P2(2) - RA2PPP2(2))
X      + ABS(RA2P2(3) - RA2PPP2(3))
  IF ((ERRA3.GT..0001.OR.ERRR2.GT..0001).AND.ITR84.LT.101)
X    GO TO 310

C Lift point coordinates from load cg
  DO 370 J = 1,NC
    R2SJ2(1,J) = R2PJ2(1,J) - R2P2S2(1)
    R2SJ2(2,J) = R2PJ2(2,J) - R2P2S2(2)
370  R2SJ2(3,J) = R2PJ2(3,J) - R2P2S2(3)

C Residual load angular acceln
  SUMSM2(1) = ( FC22(2)*RA2S2(3)-RA2S2(2)*FC22(3)+MA22(1))/I2XX
  SUMSM2(2) = (-FC22(1)*RA2S2(3)+RA2S2(1)*FC22(3)+MA22(2))/I2YY
  SUMSM2(3) = ( MA22(3)+FC22(1)*RA2S2(2)-RA2S2(1)*FC22(2))/I2ZZ

C Cable force identity error
  SUMFC22(1) = FC22(1)
  SUMFC22(2) = FC22(2)
  SUMFC22(3) = FC22(3)
  DO 380 J = 1,NC
    TAUJ(J) = KS*(LCJ(J) - LCJO(J))
    IF (TAUJ(J).LT.0.) TYPE *,
X    'STRETCH TRIM FAILED (NEG TAU) ... CHECK LOAD CG PARAMETERS'
    SUMFC22(1) = SUMFC22(1) + TAUJ(J)*KCJ2(1,J)
    SUMFC22(2) = SUMFC22(2) + TAUJ(J)*KCJ2(2,J)
380  SUMFC22(3) = SUMFC22(3) + TAUJ(J)*KCJ2(3,J)

C pht added to set final value of ra2*2 before returning to ghslmc
  RA2S2(1) = RA2P2(1) + R2P2S2(1)
  RA2S2(2) = RA2P2(2) + R2P2S2(2)
  RA2S2(3) = RA2P2(3) + R2P2S2(3)

C*** END OF ELASTIC SLING TRIM
400  CONTINUE
      RETURN
      END

```

## B.8. SUBROUTINE NRT\_UNC3\_IN(NPTS)

*Functions:* Reads FORTRAN binary format control history into Gen Hel non-real-time run array XMINPUT. Records number of data points in the input file as NPTS.

*Inputs:* CHFILE, AXIS

*Outputs:* XMINPUT, NPTS

*Called By:* bhawk\_nrt\_io

*Calls:* None

*Comments:* Control history file can be either from actual flight data or from makesweep. In either case, file must contain 5 channels: TIME, DA, DB, DP, DC. For the computer generated sweep, the on-axis channel is selected by the variable AXIS, where: (1) is DA, lateral axis; (2) is DB, longitudinal axis; (3) is DP, yaw axis; and (4) is DC, heave axis.

### Program Listing:

```
C nrt_unc3_in.f ----- 31 AUG 98 Peter Tyson

C Reads UNC3 formatted control history into GenHel non-real-
C time run array XMINPUT. Control history file must contain 5
C records: time, da, db, dp, dc. If computer generated MAKESWEEP
C is used, records are time, on-axis input, 3 off-axis (noise).

      SUBROUTINE NRT_UNC3_IN(NPTS)

      INCLUDE 'nrt_param.cmn'
      INCLUDE 'slvars.cmn'

      CHARACTER*40 FN
      INTEGER J(5)

      COMMON /XMASIN/ XMINPUT(NMASR,MAXDIM)

      AXSP1 = AXIS + 1
      AXSP2 = AXIS + 2
      AXSP3 = AXIS + 3
      IF (AXSP1.GT.4) AXSP1 = AXSP1 - 4
      IF (AXSP2.GT.4) AXSP2 = AXSP2 - 4
      IF (AXSP3.GT.4) AXSP3 = AXSP3 - 4

      J(1) = 1
      J(2) = AXIS + 1
      J(3) = AXSP1 + 1
      J(4) = AXSP2 + 1
      J(5) = AXSP3 + 1

C--- read in control history file if first time ---C
```

```

      IF (NREC .EQ. 0) THEN
        NLAST = LASTCHR(CHFILE)
        FN = CHFILE(1:NLAST)
        WRITE(6,210) FN
        OPEN(201,FILE=FN,FORM='unformatted',STATUS='old')
        DO I = 1,MAXDIM
          READ(201,ERR=100) (XMINPUT(J(K),I),K=1,5)
          NREC = I
        END DO
100    WRITE(6,211) I-1
        CLOSE(201)
      END IF
      NPTS = NREC

      RETURN

210  FORMAT(//5X,'READING CONTROL HISTORY FROM 'A)
211  FORMAT( /5X,'NUMBER OF STORES READ    = 'I8)

      END

```

## B.9. SUBROUTINE `nrt_unc3_out(NRESET)`

**Functions:** Writes data to FORTRAN binary data file(s) for post-run processing into UNC3 format data files for use in CIPHER® and XPLOT. Output files are tailored to the helicopter/load configuration being analyzed: no-load runs, multi-cable, and single-cable configurations store different data as output.

**Inputs:** DATAFILE, NGAJFPS, slung load variables in `slvars.cmn`

**Outputs:** `datafile.dat` contains trim values and simulation run data  
`datafile.da2` containing main rotor wake data

**Called By:** `bhawk_nrt_io`

**Calls:** None

**Comments:** Initial call or first call after reset writes the trim values to the data output file and opens the wake data file (if required). Subsequent passes through the program write a set of simulation values at each time step. Wake data is written to `datafile.da2` by the subroutine `wake`.

### Program Listing:

```
C nrt_unc3_out.f ----- 31 AUG 98 Peter Tyson

C   Writes formatted output to binary file for GenHel/Slung Load
C   simulation. Output file 'name.dat' is then read and reformatted
C   by ghs1_dat.f (runghsldat) to create 'name.out' (ASCII header file)
C   and 'name.xp' (UNC3 format XPLOT/CIPHER file).
C   This subroutine is called by BHAWK_NRT_IO in the non-real-time
C   run of GenHel, version 6.0. NRESET true opens output files and
C   writes header information. NREC contains the number of records read
C   from the input file (done after first time through NRT_UNC3_OUT
C   and so is written to data file on second pass)

      SUBROUTINE NRT_UNC3_OUT(NRESET)

      INCLUDE 'slvars.cmn'

      COMMON /ISCASC/ ISCAS(50)
      EQUIVALENCE (ISCAS(6), NGAJFPS)
      EQUIVALENCE (A(148), FNORTH), (A(149), FE), (A(150), FD)
      EQUIVALENCE (A(164), TTL), (A(165), TTM), (A(166), TTN)
      EQUIVALENCE (FCS(67), XAT), (FCS(68), XBT), (FCS(69), XPT)

      CHARACTER*40 FN, FN2
      INTEGER FIRSTPASS
      DATA FIRSTPASS /0/

C--- first pass after reset, open files, write header info, return ---C
```

```

IF (NRESET .NE. 0) THEN
  NRESET = 0
  FIRSTPASS = 1
  NLAST = LASTCHR(DATAFILE)
  FN = DATAFILE(1:NLAST)//".dat"
  OPEN(1,FILE=FN,FORM='UNFORMATTED',STATUS='UNKNOWN')
  WRITE(1) NSTORE, DT, ILOAD, IWAKE, ISWIRL, IPILOT, IDATA,
X   NGAJFPS, TOW, FWT, W1, I1XX, I1YY, I1ZZ, I1XZ, AXIS, R1SN,
X   PH1DEG, TH1DEG, PS1DEG, (A(J),J=64,66), PSVA, FA11, MA11,
X   XAAD, XBAD, XCAD, XPAD
  WRITE(1) CHFILE

  IF (ILOAD.EQ.0) WRITE(6,17) FN

  IF (ILOAD.EQ.1) THEN
    WRITE(6,18) FN
    WRITE(1) LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ, R1SA1,
X   R2S12, DOQ, STRETCH, LCO, LC, KS, CS, QO, V1S1,
X   FA1N, FA2N, PH2*R2D, TH2*R2D, PS2*R2D, PHC, THC, Q(12),
X   (U(j),j=1,12)
  END IF

  IF (ILOAD.EQ.2) THEN
    WRITE(6,19) FN
    WRITE(1) LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ, STRETCH,
X   IAERSL, NC, R2SJ2, RA2PO2, R2P2SO2, R2P2S2, KS, CS,
X   R1SA1, R2PJ2, DOQ, GAM*RTD, PSVA*RTD, ALF2D, BET2D,
X   DETG, LCJO, LCJ, TAUJ, FA1N, FC11, FC1N, MC11, FA22,
X   FA2N, MA22, (A(J),J=64,66), PH2*R2D, TH2*R2D, PS2*R2D,
X   RA2S2, VA2SN, R1S2SN, DYNAMIC
  END IF

  IF ((ILOAD.NE.0).AND.(IWAKE.EQ.1)) THEN
    FN2 = DATAFILE(1:NLAST)//".da2"
    OPEN(3,FILE=FN2,FORM='UNFORMATTED',STATUS='UNKNOWN')
    WRITE(3) NSTORE
    WRITE(6,20) FN2
  END IF
  RETURN
END IF

C--- write number of input data points on second pass only ---C
IF (FIRSTPASS.NE.0) THEN
  WRITE(1) NREC
  FIRSTPASS = 0
END IF

C--- write data each time called ---C
IF (ILOAD.EQ.0) WRITE(1) T, (A(j),j=88,90), (A(j),j=55,57),
X (A(j),j=64,66), (A(j),j=37,39), (A(j),j=4,9),
X A(106), A(107), A(176), FA11, MA11, XAAD, XBAD, XCAD, XPAD,
X RSAS, PSAS, YSAS, DMIXA, DMIXB, DMIXC, DMIXP, PSFWD, PSAFT,
X PSLAT, PSTR

```

```

      IF (ILOAD.EQ.1) WRITE(1) T, DU, TAU, U, (DQ(I),I=4,11), Q,
X    R2SN, V2SN, DV2SN, RSN, VSN, DELRSN, DELVSN, DELDVSN, SSFN,
X    ISSFN, IISSFN, FA11, FA1N, FA2N, MA11, MA22, OMCC, LCMLCO,
X    RHK2S1, FC11, MC11, XAAD, XBAD, XCAD, XPAD, PHCH, THCH,
X    RPHC, RTHC, RSAS, PSAS, YSAS, DMIXA, DMIXB, DMIXC, DMIXP,
X    PSFWD, PSAFT, PSLAT, PSTR

      IF ((ILOAD.EQ.2).AND.(IDATA.EQ.1)) WRITE(1) T, DU, U,
X    (DQ(I),I=4,9), Q, P2P, Q2P, FC11, FC1N, MC11, FA11, FA1N,
X    MA11, FA22, FA2N, MA22, FA2W, MA2W, V2SN, VA2S2, RA2S1,
X    R1S2SN, XAAD, XBAD, XCAD, XPAD, RSAS, PSAS, YSAS, DMIXA,
X    DMIXB, DMIXC, DMIXP, PSFWD, PSAFT, PSLAT, PSTR, float(QUAD),
X    ALF2D, BET2D, DALF2D, DBET2D, DANG2D, P21, Q21, R21, P2N, Q2N,
X    R2N

      IF ((ILOAD.EQ.2).AND.(IDATA.EQ.0)) WRITE(1) T, (U(I),I=4,6),
X    XAAD, XBAD, XCAD, XPAD, P2P, Q2P, RSAS, PSAS, YSAS, DMIXA,
X    DMIXB, DMIXC, DMIXP

      RETURN
15  FORMAT(/)
17  FORMAT(/5x'WRITING NO-LOAD HEADER TO '16A)
18  FORMAT(/5x'WRITING SL SINGLE CABLE HEADER TO '16A)
19  FORMAT(/5x'WRITING SL MULTI-CABLE HEADER TO '16A)
20  FORMAT(/5x'WRITING WAKE INFORMATION TO '16A)
      END

```

## B.10. SUBROUTINE **pilot(NRESET)**

*Functions:* Prevents the un-piloted simulation from becoming unstable when run from a control history input file (either actual flight test data or from a computer generated sweep).

*Inputs:* PHI, THE, PSI, P, Q, R, XAAD, XBAD, XPAD, XCAD, VX, VY, VZ

*Outputs:* XAAD, XBAD, XPAD, XCAD, NRESET

*Called By:* nrt\_master\_read

*Calls:* None

*Comments:* On the first pass through the routine in Pilot Input mode (after control input file has been read such that NREC > 0), the current control positions, helicopter body Euler angles, and helicopter velocities are recorded as the trim values for use in the feedback loop. Gain values used for attitude and rate feedback are from Ref [27]. Velocity feedback for the off-axis cyclic controls is added to help maintain trim position and airspeed.

### *Program Listing:*

C pilot.f ----- 12 AUG 98 Peter Tyson

```
SUBROUTINE PILOT(NRESET)

  INCLUDE 'slvars.cmn'

  INTEGER FIRST

  EQUIVALENCE (A( 58), VX)
  EQUIVALENCE (A( 59), VY)
  EQUIVALENCE (A( 60), VZ)

  DATA KPHI, KP      /10.0, 1.0/,
X      KTHE, KQ      / 5.0, 1.0/,
X      KPSI, KR      / 2.0, 1.0/,
X      K VX, K VY, K VZ / 0.0, 0.0, 0.0/,
X      FIRST        /1/

  IF (IPILOT.NE.1) RETURN

  IF ((NREC.NE.0).AND.(FIRST.EQ.1)) THEN
    FIRST = 0
    NRESET = 0
    PHIREF = A(4)
    THEREF = A(5)
    PSIREF = A(6)
    XAREF = XAAD
    XBREF = XBAD
```

```

      XPREF = XPAD
      XCREF = XCAD
      VXREF = VX
      VYREF = VY
      VZREF = VZ
END IF

XAAD= XAAD+(XAAD-XAREF)/10.- (A(4)-PHIREF)*KPHI- A(7)*KP
XBAD= XBAD+(XBAD-XBREF)/10.- (A(5)-THEREF)*KTHE- A(8)*KQ
XPAD= XPAD+(XPAD-XPREF)/10.- (A(6)-PSIREF)*KPSI- A(9)*KR
XCAD= XCAD-(XCAD-XCREF)/10.+ (VZ-VZREF)*KVZ

IF (AXIS.EQ.1) THEN
      XBAD = XBAD+(VX-VXREF)*KVX
ELSE IF (AXIS.EQ.2) THEN
      XAAD = XAAD-(VY-VYREF)*KVY
ELSE
      XAAD = XAAD-(VY-VYREF)*KVY
      XBAD = XBAD+(VX-VXREF)*KVX
END IF

RETURN
END

```



## B.11. SUBROUTINE *wake*

*Functions:* Compute the wind velocity at the load center of gravity due to the main rotor downwash. Add this value to the relative wind due to load motion through space. Output wake parameters to a data file at each time step to allow for post-run analysis.

*Inputs:* A1F, B1F, XIS, XMUXS, XMUYS, XMUZS, PH1, TH1, PS1, slung load variables in *slvars.cmn*

*Outputs:* VA2S2 relative air velocity at the load center of gravity due to load translational motion and rotordownwash model

*Called By:* ghslmc, ghslmc\_ic

*Calls:* None

*Comments:* Wake geometry is computed through momentum theory and with variables provided from Gen Hel's rotor subroutine. Load c.g. position relative to the main rotor hub is computed in wake coordinates, and then corrected for wake offset and elliptical wake contraction. Load position then determines axial and tangential velocity components. Tangential velocity direction is computed based upon load position relative to center of wake core, assuming a counterclockwise (from above) wake rotation.

### *Program Listing:*

```
C wake.f ----- 03 SEP 98 Peter Tyson

C Called by GHSLMC, or GHSLMC_IC, running the
C GenHel/Slung Load non-real-time simulation.

      SUBROUTINE WAKE

      INCLUDE 'slvars.cmn'

      REAL RHH1(3), RHHN(3), ZR(4), WAKEDATA(4,51),
X      T1N(3,3), TN1(3,3), TWN(3,3), TNW(3,3),
X      V1(3), VO1(3), VPP1(3), VPPN(3), VP1(3), VPN(3), VPW(3),
X      RA2SN(3), RH2SN(3), RH2SW(3), RH2SWO(3),
X      RADIUS, HEIGHT, DL, SIGMA,
X      ROW, COL, LO, HI, VWW(3), VWN(3), VW2(3), VA2S2W(3)

      DATA RHH1, XIS /-11.4, 0., 119.5, -3.0/
      DATA ZR /
X      0.3260,      0.6600,      0.9930,      1.3260/

C dynamic pressure over disc loading, q/DL
```

```

DATA WAKEDATA /
x  -0.0001,  0.1633,  0.4546,  0.5284,
x  -0.0001,  0.1658,  0.4558,  0.5286,
x   0.0002,  0.1727,  0.4599,  0.5290,
x   0.0014,  0.1813,  0.4673,  0.5293,
x   0.0039,  0.1890,  0.4758,  0.5318,
x   0.0089,  0.1952,  0.4824,  0.5422,
x   0.0174,  0.2009,  0.4889,  0.5594,
x   0.0286,  0.2098,  0.4992,  0.5718,
x   0.0410,  0.2248,  0.5116,  0.5738,
x   0.0554,  0.2418,  0.5222,  0.5743,
x   0.0731,  0.2571,  0.5309,  0.5810,
x   0.0953,  0.2746,  0.5401,  0.5943,
x   0.1242,  0.2984,  0.5520,  0.6137,
x   0.1602,  0.3264,  0.5683,  0.6423,
x   0.2007,  0.3557,  0.5877,  0.6805,
x   0.2426,  0.3851,  0.6059,  0.7200,
x   0.2857,  0.4161,  0.6227,  0.7530,
x   0.3319,  0.4577,  0.6438,  0.7783,
x   0.3810,  0.5157,  0.6781,  0.7965,
x   0.4298,  0.5760,  0.7271,  0.8094,
x   0.4750,  0.6250,  0.7784,  0.8199,
x   0.5157,  0.6718,  0.8213,  0.8307,
x   0.5537,  0.7260,  0.8566,  0.8438,
x   0.5934,  0.7763,  0.8866,  0.8609,
x   0.6389,  0.8127,  0.9098,  0.8806,
x   0.6919,  0.8489,  0.9255,  0.8942,
x   0.7480,  0.8992,  0.9385,  0.8956,
x   0.8006,  0.9560,  0.9539,  0.8902,
x   0.8473,  1.0078,  0.9667,  0.8838,
x   0.8904,  1.0509,  0.9660,  0.8706,
x   0.9361,  1.0862,  0.9518,  0.8448,
x   0.9917,  1.1235,  0.9354,  0.8140,
x   1.0614,  1.1688,  0.9262,  0.7840,
x   1.1357,  1.2021,  0.9207,  0.7459,
x   1.1914,  1.2006,  0.9010,  0.6912,
x   1.2312,  1.1572,  0.8499,  0.6214,
x   1.2865,  1.0680,  0.7660,  0.5422,
x   1.3255,  0.9266,  0.6587,  0.4673,
x   1.2585,  0.7380,  0.5435,  0.4068,
x   1.0855,  0.5570,  0.4417,  0.3526,
x   0.8648,  0.4324,  0.3635,  0.2956,
x   0.5962,  0.3394,  0.3011,  0.2399,
x   0.3100,  0.2467,  0.2445,  0.1926,
x   0.1082,  0.1710,  0.1907,  0.,
x   0.0306,  0.1317,  0.1424,  0.,
x   0.0224,  0.1123,  0.1042,  0.,
x   0.0238,  0.0935,  0., 0.,
x   0.0183,  0.0829,  0., 0.,
x   0.0130,  0., 0., 0.,
x   4*0.0,
x   4*0.0/

```

C position kinematics: read T1N, TN1 from strike, compute T2N

```

DO 101 I = 1,3
DO 101 J = 1,3
T1N(I,J) = A(15+I+(J-1)*3)
101 TN1(J,I) = A(15+I+(J-1)*3)
SPH2      = SIN(PH2)
CPH2      = COS(PH2)
STH2      = SIN(TH2)
CTH2      = COS(TH2)
SPS2      = SIN(PS2)
CPS2      = COS(PS2)
T2N(1,1)  = CPS2*CTH2
T2N(1,2)  = CTH2*SPS2
T2N(1,3)  = -STH2
T2N(2,1)  = CPS2*SPH2*STH2-CPH2*SPS2
T2N(2,2)  = SPH2*SPS2*STH2+CPH2*CPS2
T2N(2,3)  = CTH2*SPH2
T2N(3,1)  = CPH2*CPS2*STH2+SPH2*SPS2
T2N(3,2)  = CPH2*SPS2*STH2-CPS2*SPH2
T2N(3,3)  = CPH2*CTH2
TN2(1,1)  = T2N(1,1)
TN2(1,2)  = T2N(2,1)
TN2(1,3)  = T2N(3,1)
TN2(2,1)  = T2N(1,2)
TN2(2,2)  = T2N(2,2)
TN2(2,3)  = T2N(3,2)
TN2(3,1)  = T2N(1,3)
TN2(3,2)  = T2N(2,3)
TN2(3,3)  = T2N(3,3)

```

C Euler angles, Tip Path Plane (TPP) to HC body axes

```

SPHT = SIN(B1F/R2D)
CPHT = COS(B1F/R2D)
STHT = SIN((XIS+A1F)/R2D)
CTHT = COS((XIS+A1F)/R2D)

```

C Relative wind at hub from hub translational velocity in HC body axes

```

SXIS = SIN(XIS/R2D)
CXIS = COS(XIS/R2D)
V1(1) = - OMEGA*RMR*(CXIS*XMUXS + SXIS*XMUZS)
V1(2) = - OMEGA*RMR*XMUYS
V1(3) = - OMEGA*RMR*(CXIS*XMUZS - SXIS*XMUXS)

```

C Rotor downwash: normalized inflow \* tip speed, in TPP, HC body axes

```

VO = ABS((XLAMDA - XMUYS)*OMEGA*RMR)
VO1(1) = CPHT*STHT*VO
VO1(2) = - SPHT *VO
VO1(3) = CPHT*CTHT*VO

```

C V'' (far-wake) in HC body, Inertial axes

```

VPP1(1) = V1(1) + 2*VO1(1)
VPP1(2) = V1(2) + 2*VO1(2)
VPP1(3) = V1(3) + 2*VO1(3)
VPPN(1) = TN1(1,3)*VPP1(3)+TN1(1,2)*VPP1(2)+VPP1(1)*TN1(1,1)
VPPN(2) = TN1(2,3)*VPP1(3)+VPP1(2)*TN1(2,2)+VPP1(1)*TN1(2,1)

```

```

      VPPN(3) = VPP1(3)*TN1(3,3)+VPP1(2)*TN1(3,2)+VPP1(1)*TN1(3,1)

C   Total V'' and Euler wake angles
      VPP = SQRT(VPPN(1)**2 + VPPN(2)**2 + VPPN(3)**2)
      PHIW = ASIN(-VPPN(2)/VPP)
      THEW = ATAN2(VPPN(1),VPPN(3))

C   position kinematics: compute TWN, TNW
      SPHW = SIN(PHIW)
      CPHW = COS(PHIW)
      STHW = SIN(THEW)
      CTHW = COS(THEW)
      TNW(1,1) = CTHW
      TNW(1,2) = SPHW*STHW
      TNW(1,3) = CPHW*STHW
      TNW(2,1) = 0.
      TNW(2,2) = CPHW
      TNW(2,3) = -SPHW
      TNW(3,1) = -STHW
      TNW(3,2) = CTHW*SPHW
      TNW(3,3) = CPHW*CTHW
      TWN(1,1) = TNW(1,1)
      TWN(1,2) = TNW(2,1)
      TWN(1,3) = TNW(3,1)
      TWN(2,1) = TNW(1,2)
      TWN(2,2) = TNW(2,2)
      TWN(2,3) = TNW(3,2)
      TWN(3,1) = TNW(1,3)
      TWN(3,2) = TNW(2,3)
      TWN(3,3) = TNW(3,3)

C   V' (near-wake) in HC body, Inertial, Wake axes
      VP1(1) = V1(1) + VO1(1)
      VP1(2) = V1(2) + VO1(2)
      VP1(3) = V1(3) + VO1(3)
      VPN(1) = TN1(1,3)*VP1(3)+TN1(1,2)*VP1(2)+VP1(1)*TN1(1,1)
      VPN(2) = TN1(2,3)*VP1(3)+VP1(2)*TN1(2,2)+VP1(1)*TN1(2,1)
      VPN(3) = VP1(3)*TN1(3,3)+VP1(2)*TN1(3,2)+VP1(1)*TN1(3,1)
      VPW(1) = TWN(1,3)*VPN(3)+TWN(1,2)*VPN(2)+VPN(1)*TWN(1,1)
      VPW(2) = TWN(2,3)*VPN(3)+VPN(2)*TWN(2,2)+VPN(1)*TWN(2,1)
      VPW(3) = VPN(3)*TWN(3,3)+VPN(2)*TWN(3,2)+VPN(1)*TWN(3,1)
      VP = SQRT(VPN(1)**2 + VPN(2)**2 + VPN(3)**2)

C   far-wake center offset
C   xcw = (u'w/2) * 1.5*R/((V'+V'')/2) = u'w*R/(2*vo)
C   ycw = (v'w/2) * 1.5*R/((V'+V'')/2) = v'w*R/(2*vo)
      XCW = VPW(1)*RMR/(2*VO)
      YCW = VPW(2)*RMR/(2*VO)

C   load c.g. distance from rotor hub, HC body, Inertial, Wake axes
C   IC mode, recompute load posit each time fm ra2*2 = ra2'o2 + r2'2*o2
      IF (IMODE.LT.0) THEN
        RA2SN(1)=TN2(1,3)*RA2S2(3)+TN2(1,2)*RA2S2(2)+RA2S2(1)*TN2(1,1)
        RA2SN(2)=TN2(2,3)*RA2S2(3)+RA2S2(2)*TN2(2,2)+RA2S2(1)*TN2(2,1)

```

```

      RA2SN(3)=RA2S2(3)*TN2(3,3)+RA2S2(2)*TN2(3,2)+RA2S2(1)*TN2(3,1)
      END IF
      RHHN(1) = TN1(1,3)*RHH1(3)+TN1(1,2)*RHH1(2)+RHH1(1)*TN1(1,1)
      RHHN(2) = TN1(2,3)*RHH1(3)+RHH1(2)*TN1(2,2)+RHH1(1)*TN1(2,1)
      RHHN(3) = RHH1(3)*TN1(3,3)+RHH1(2)*TN1(3,2)+RHH1(1)*TN1(3,1)
      RH2SN(1) = RA2SN(1) + RHHN(1)/12
      RH2SN(2) = RA2SN(2) + RHHN(2)/12
      RH2SN(3) = RA2SN(3) + RHHN(3)/12
      RH2SW(1) = TWN(1,3)*RH2SN(3)+TWN(1,2)*RH2SN(2)+RH2SN(1)*TWN(1,1)
      RH2SW(2) = TWN(2,3)*RH2SN(3)+RH2SN(2)*TWN(2,2)+RH2SN(1)*TWN(2,1)
      RH2SW(3) = RH2SN(3)*TWN(3,3)+RH2SN(2)*TWN(3,2)+RH2SN(1)*TWN(3,1)

C   load distance from rotor hub, wake axes, with wake center offset
      RH2SWO(1) = RH2SW(1) + XCW
      RH2SWO(2) = RH2SW(2) + YCW
      RH2SWO(3) = RH2SW(3)

C   load nondimensional distance from hub, w/ elliptical wake correction
      STHETAP = ABS(COS(B1F/R2D)*COS(A1F/R2D)*XLAMDA*OMEGA*RMR)/VP
      IF (STHETAP.GT.1.) STHETAP = 1.
      THETAP = ASIN(STHETAP)*R2D
      RMIN = RMR*STHETAP
      TPSIP = RH2SWO(1)/RH2SWO(2)
      SQYP = (RMR**2 * RMIN**2)/(RMIN**2 + TPSIP**2*RMR**2)
      SQZP = TPSIP**2*SQYP
      REP = SQRT(SQYP + SQZP)
      RP = SQRT(RH2SWO(1)**2 + RH2SWO(2)**2)
      RADIUS = RP/REP
      RAD = RP/RMR
      HEIGHT = RH2SW(3)/RMR

C   wake velocity from wake data (NACA TN 4239)
      DL = CTA*RHO*(OMEGA*RMR)**2
      SIGMA = 4*1.73*R2D/180./RMR
      VWZ = 0.
      IF (RADIUS.LE.1) THEN
        ROW = INT(RADIUS/0.02) + 1
        COL = 1
        IF (HEIGHT.GE.ZR(2)) COL = 2
        IF (HEIGHT.GE.ZR(3)) COL = 3
        LO = WAKEDATA(COL, ROW)
        HI = WAKEDATA(COL+1,ROW)
        QZ = DL*(LO+(HI-LO)*(HEIGHT-ZR(COL))/(ZR(COL+1)-ZR(COL)))
        VWZ = SQRT(2*QZ/RHO)
      END IF

C   tangential velocity (USAAMRDL 72-33 Fig 24) and direction, Wake axes
      VWT = 0.1 * SQRT(DL/2/RHO)
      IF (RADIUS.LT.0.4) VWT = VWT*(RADIUS/0.4)
      IF (RADIUS.GE.1.0) VWT = 0.
      PST = 90./R2D
      IF (RADIUS.NE.0.) PST = ATAN2(RH2SWO(2),RH2SWO(1)) - 90./R2D

C   wake velocity from table: wake axes, inertial, then load body axes

```

```

VWW(1) = 0.
VWW(2) = 0.
VWW(3) = VWZ
IF (ISWIRL.NE.0) THEN
  VWW(1) = VWT*COS(PST)
  VWW(2) = VWT*SIN(PST)
END IF
VWN(1) = TNW(1,3)*VWW(3)+TNW(1,2)*VWW(2)+VWW(1)*TNW(1,1)
VWN(2) = TNW(2,3)*VWW(3)+VWW(2)*TNW(2,2)+VWW(1)*TNW(2,1)
VWN(3) = VWW(3)*TNW(3,3)+VWW(2)*TNW(3,2)+VWW(1)*TNW(3,1)
VW2(1) = T2N(1,3)*VWN(3)+T2N(1,2)*VWN(2)+VWN(1)*T2N(1,1)
VW2(2) = T2N(2,3)*VWN(3)+VWN(2)*T2N(2,2)+VWN(1)*T2N(2,1)
VW2(3) = VWN(3)*T2N(3,3)+VWN(2)*T2N(3,2)+VWN(1)*T2N(3,1)

```

C load velocity WRT airmass, including wake, load body axes

```

VA2S2W(1) = VA2S2(1) - VW2(1)
VA2S2W(2) = VA2S2(2) - VW2(2)
VA2S2W(3) = VA2S2(3) - VW2(3)

```

C wake data printout to .da2 file

```

IF ((IMODE.GT.0).AND.(IDATA.EQ.0)) WRITE(3) T, A1F, B1F, VO,
X VPP, VP, PHIW*R2D, THEW*R2D, XCW, YCW, RH2SWO, THETAP,
X RADIUS, HEIGHT, VWZ, PST*R2D, VA2S2W

```

```

IF ((IMODE.GT.0).AND.(IDATA.EQ.1)) WRITE(3) T, A1F, B1F, V1, VO,
X VO1, VPP1, VPPN, VPP, VP1, VPN, VP, PHIW*R2D, THEW*R2D, XCW,
X YCW, RH2SW, RH2SWO, THETAP, RADIUS, RAD, HEIGHT, VWZ, VWT,
X PST*R2D, VWW, VW2, VA2S2W, VA2S2

```

```

VA2S2(1) = VA2S2W(1)
VA2S2(2) = VA2S2W(2)
VA2S2(3) = VA2S2W(3)
VA2SN(1) = TN2(1,3)*VA2S2(3)+TN2(1,2)*VA2S2(2)+VA2S2(1)*TN2(1,1)
VA2SN(2) = TN2(2,3)*VA2S2(3)+VA2S2(2)*TN2(2,2)+VA2S2(1)*TN2(2,1)
VA2SN(3) = VA2S2(3)*TN2(3,3)+VA2S2(2)*TN2(3,2)+VA2S2(1)*TN2(3,1)

```

```

RETURN
END

```

## B.12. COMMON FILE `slvars.cmn`

*Functions:* Provide a common area to list and store the variables used by the slung load subroutines.

*Included By:* `conexaero`, `ghsl_init`, `ghslmc`, `ghslmc_ic`, `nrt_unc3_in`,  
`nrt_unc3_out`, `pilot`, `wake`

*Comments:* Variables used by the slung load subroutines that are found in the Gen Hel common file `model.cmn` are included as equivalences in `slvars.cmn`.

### *Program Listing:*

C `slvars.cmn` ----- 31 AUG 98 Peter Tyson

C Common blocks for the GenHel/Slung Load Simulation.

C Includes variables required for data output from the No-Load,

C SLSC, and SLMC cases.

REAL

```
X PH2DEG, TH2DEG, PS2DEG, ALF2D, BET2D, OM2N(3), V2SN(3),
X VA2SN(3), VA2S2(3), FA11(3), FA1N(3), MA11(3), FC11(3),
X FC1N(3), MC11(3), FA22(3), FA2N(3), MA22(3), FC22(3),
X FC2N(3), MC22(3), FA2W(3), MA2W(3), W2, I2XX, I2YY, I2ZZ,
X I2XZ, KS, CS, DOQ, LCJ(8), LCJO(8), TAUJ(8), R2SJ2(3,8),
X RA2P2(3), R2P2S2(3), RA2PO2(3), R2P2SO2(3), R2PJ2(3,8),
X TAU, RSN(3), DELRSN(3), R2SN(3), VSN(3), DELVSN(3),
X DV2SN(3), DELDVSN(3), SSFN(3), ISSFN(3), IISSFN(3),
X OMCC(3), LCO, LCMLCO, PHCH, THCH, RPHC, RTHC, R2S12(3),
X RA2S1(3), TOW, FWT, GAM, PSVA,
X DETG, R1SA1(3), DELPS2O, ALFO, PSWO, FREQ, V1S1(3),
X QO(12), DU(12), U(12), DQ(12), Q(12), DV1SN(3), DOM11(3),
X DOM22(3), DDRA2S2(3), OM11(3), OM22(3), DRA2S2(3), DPH1,
X DTH1, DPS1, DPH2, DTH2, DPS2, R1SN(3), PH1, TH1, PH2, TH2,
X PS2, RA2S2(3), PH1DEG, TH1DEG, PS1DEG, WN(3), G, I1XX,
X I1YY, I1ZZ, I1XZ, DT, W1, RHO, T, R2D, XMUXH, XMUYH, XMUZH,
X XMUXS, XMUYS, XMUZS, XLAMDA, CTA, OMEGA, A1F, B1F, FSCG,
X WLCG, BLCG, RMR, T2N(3,3), TN2(3,3), DALF2D, DBET2D, DANG2D,
X ALFAO, BETAO, R1S2SN(3), P2N, Q2N, R2N, P21, Q21, R21
INTEGER STRETCH, NSTORE, NREC, NC, QUAD, AXIS
COMMON /LFLOAT/ RLOAD(500)
```

C LOAD EULER ROLL, PITCH AND YAW ANGLES - DEG

EQUIVALENCE (RLOAD( 1), PH2DEG )

EQUIVALENCE (RLOAD( 2), TH2DEG )

EQUIVALENCE (RLOAD( 3), PS2DEG )

C LOAD ANGLES OF ATTACK AND SIDESLIP - DEG

EQUIVALENCE (RLOAD( 4), ALF2D )

EQUIVALENCE (RLOAD( 5), BET2D )

C LOAD ROLL, PITCH AND YAW RATES, INERTIAL FRAME - RAD/SEC

EQUIVALENCE (RLOAD( 6), OM2N(1) )

C LOAD ROLL, PITCH AND YAW RATES, LOAD-HC HEADING COORDS- RAD/SEC  
     EQUIVALENCE (RLOAD( 9), P2P )  
     EQUIVALENCE (RLOAD( 10), Q2P )

C LOAD VELOCITIES, INERTIAL FRAME - FT/SEC  
     EQUIVALENCE (RLOAD( 11), V2SN(1) )

C LOAD APPARENT WIND VELOCITIES, INERTIAL FRAME - FT/SEC  
     EQUIVALENCE (RLOAD( 14), VA2SN(1) )

C LOAD APPARENT WIND VELOCITIES, LOAD BODY FRAME - FT/SEC  
     EQUIVALENCE (RLOAD( 17), VA2S2(1) )

C HELO AERO FORCES, HC BODY AXES  
     EQUIVALENCE (RLOAD( 20), FA11(1) )

C HELO AERO FORCES, INERTIAL AXES  
     EQUIVALENCE (RLOAD( 23), FA1N(1) )

C HELO AERO MOMENTS ABOUT HC CG, HC BODY AXES  
     EQUIVALENCE (RLOAD( 26), MA11(1) )

C HOOK FORCES, HC BODY AXES  
     EQUIVALENCE (RLOAD( 29), FC11(1) )

C HOOK FORCES, INERTIAL AXES  
     EQUIVALENCE (RLOAD( 32), FC1N(1) )

C HOOK MOMENTS ABOUT HC CG, HC BODY AXES  
     EQUIVALENCE (RLOAD( 35), MC11(1) )

C LOAD AERO FORCES, LOAD BODY AXES  
     EQUIVALENCE (RLOAD( 38), FA22(1) )

C LOAD AERO FORCES, INERTIAL AXES  
     EQUIVALENCE (RLOAD( 41), FA2N(1) )

C LOAD AERO MOMENTS ABOUT LOAD CG, LOAD BODY AXES  
     EQUIVALENCE (RLOAD( 44), MA22(1) )

C SLING FORCES, LOAD BODY AXES  
     EQUIVALENCE (RLOAD( 47), FC22(1) )

C SLING FORCE, INERTIAL AXES  
     EQUIVALENCE (RLOAD( 50), FC2N(1) )

C SLING MOMENTS ABOUT LOAD CG, LOAD BODY AXES  
     EQUIVALENCE (RLOAD( 53), MC22(1) )

C LOAD AERO FORCES, LOAD APPARENT WIND AXES  
     EQUIVALENCE (RLOAD( 56), FA2W(1) )

C LOAD AERO MOMENTS ABOUT LOAD CG, LOAD APPARENT WIND AXES  
     EQUIVALENCE (RLOAD( 59), MA2W(1) )



C RLOAD( 62) - ( 69) EMPTY  
  
 C LOAD WEIGHT (POUNDS)  
     EQUIVALENCE (RLOAD( 70), W2 )  
  
 C LOAD XX, YY, ZZ, AND XZ MOMENTS OF INERTIA - SLUG-FT2  
     EQUIVALENCE (RLOAD( 71), I2XX )  
     EQUIVALENCE (RLOAD( 72), I2YY )  
     EQUIVALENCE (RLOAD( 73), I2ZZ )  
     EQUIVALENCE (RLOAD( 74), I2XZ )  
  
 C SLING SPRING CONSTANT  
     EQUIVALENCE (RLOAD( 75), KS )  
  
 C SLING DAMPING CONSTANT  
     EQUIVALENCE (RLOAD( 76), CS )  
  
 C LOAD DRAG OVER DYNAMIC PRESSURE  
     EQUIVALENCE (RLOAD( 77), DOQ )  
  
 C SLING CABLE LENGTH (1 TO 8 LEGS)  
     EQUIVALENCE (RLOAD( 78), LCJ(1) )  
  
 C SLING IC LENGTH (1 TO 8 LEGS)  
     EQUIVALENCE (RLOAD( 86), LCJO(1) )  
  
 C SLING CABLE TENSIONS  
     EQUIVALENCE (RLOAD( 94), TAUJ(1) )  
  
     EQUIVALENCE (RLOAD(103), R2SJ2(1,1) )  
     EQUIVALENCE (RLOAD(127), RA2P2(1) )  
     EQUIVALENCE (RLOAD(130), R2P2S2(1) )  
     EQUIVALENCE (RLOAD(133), RA2PO2(1) )  
     EQUIVALENCE (RLOAD(136), R2P2SO2(1) )  
     EQUIVALENCE (RLOAD(139), R2PJ2(1,1) )  
     EQUIVALENCE (RLOAD(163), NREC )  
  
 C SLING TENSION  
     EQUIVALENCE (RLOAD(164), TAU )  
  
 C SYSTEM MOTION WRT INERTIAL FRAME, FT  
     EQUIVALENCE (RLOAD(165), RSN(1) )  
  
 C CHANGE IN SYSTEM MOTION WRT INERTIAL FRAME, FT  
     EQUIVALENCE (RLOAD(168), DELRSN(1) )  
  
 C LOAD MOTION WRT INERTIAL FRAME, FT  
     EQUIVALENCE (RLOAD(171), R2SN(1) )  
  
 C SYSTEM VELOCITY WRT INERTIAL FRAME, FT/S  
     EQUIVALENCE (RLOAD(174), VSN(1) )  
  
 C CHANGE IN SYSTEM VELOCITY WRT INERTIAL FRAME, FT/S  
     EQUIVALENCE (RLOAD(177), DELVSN(1) )

C LOAD ACCELERATION WRT INERTIAL FRAME, FT/S\*\*2  
EQUIVALENCE (RLOAD(180), DV2SN(1) )

C CHANGE IN SYSTEM ACCELERATION WRT INERTIAL FRAME, FT/S\*\*2  
EQUIVALENCE (RLOAD(183), DELDVS(1) )

EQUIVALENCE (RLOAD(186), SSFN(1) )  
EQUIVALENCE (RLOAD(189), ISSFN(1) )  
EQUIVALENCE (RLOAD(192), IISSFN(1) )  
EQUIVALENCE (RLOAD(195), OMCC(1) )  
EQUIVALENCE (RLOAD(198), LCO )  
EQUIVALENCE (RLOAD(200), LCMLCO )  
EQUIVALENCE (RLOAD(201), PHCH )  
EQUIVALENCE (RLOAD(202), THCH )  
EQUIVALENCE (RLOAD(203), RPHC )  
EQUIVALENCE (RLOAD(204), RTHC )

C LOCATION VECTOR, FROM HC CG TO LOAD CG, INERTIAL AXES  
EQUIVALENCE (RLOAD(205), R1S2SN(1) )

C LOCATION VECTOR  
EQUIVALENCE (RLOAD(240), R2S12(1) )

C LOCATION VECTOR, LOAD C.G. FROM HC HOOK, HC BODY AXES  
EQUIVALENCE (RLOAD(243), RA2S1(1) )

C HELICOPTER TAKEOFF WEIGHT  
EQUIVALENCE (RLOAD(246), TOW )

C HELICOPTER TAKEOFF FUEL WEIGHT  
EQUIVALENCE (RLOAD(247), FWT )

EQUIVALENCE (RLOAD(248), GAM )  
EQUIVALENCE (RLOAD(249), PSVA )  
EQUIVALENCE (RLOAD(250), DETG )  
EQUIVALENCE (RLOAD(251), R1SA1(1) )  
EQUIVALENCE (RLOAD(254), DELPS20 )  
EQUIVALENCE (RLOAD(255), ALFO )  
EQUIVALENCE (RLOAD(256), PSWO )  
EQUIVALENCE (RLOAD(257), FREQ )

C DYNAMIC AERODYNAMICS PROPORTIONALITY FACTOR  
EQUIVALENCE (RLOAD(258), DYNAMIC )

C ANGLE OF ATTACK, SIDESLIP AND TOTAL ANGULAR RATE OF CHANGE  
EQUIVALENCE (RLOAD(259), DALF2D )  
EQUIVALENCE (RLOAD(260), DBET2D )  
EQUIVALENCE (RLOAD(261), DANG2D )

C FORCE ON LOAD DUE TO ANGULAR RATES OF CHANGE  
EQUIVALENCE (RLOAD(262), FDALF2D )  
EQUIVALENCE (RLOAD(263), FDBET2D )  
EQUIVALENCE (RLOAD(264), FDANG2D )

```

C QUADRANT 1 ANGLE OF ATTACK AND SIDESLIP FOR LOAD
  EQUIVALENCE (RLOAD(265), ALFAO )
  EQUIVALENCE (RLOAD(266), BETAO )

C LOAD MOTION, INERTIAL AXES
  EQUIVALENCE (RLOAD(267), P2N )
  EQUIVALENCE (RLOAD(268), Q2N )
  EQUIVALENCE (RLOAD(269), R2N )
C LOAD MOTION, HC BODY AXES
  EQUIVALENCE (RLOAD(270), P21 )
  EQUIVALENCE (RLOAD(271), Q21 )
  EQUIVALENCE (RLOAD(272), R21 )

C RLOAD(273) - (285) EMPTY

  EQUIVALENCE (RLOAD(286), V1S1(1) )
  EQUIVALENCE (RLOAD(289), QO(1) )

C   state array equivalences for integration
C   du = {dv1sn(3),   dom11(3),   dom22(3),   ddra2s2(3)}
C   u   = { v1sn(3),   om11(3),   om22(3),   dra2s2(3)}
C   q   = { r1sn(3), ph1,th1,ps1, ph2,th2,ps2,   ra2s2(3)}
  EQUIVALENCE (RLOAD(301), DU(12) )
  EQUIVALENCE (RLOAD(321), U(12) )
  EQUIVALENCE (RLOAD(341), DQ(12) )
  EQUIVALENCE (RLOAD(361), Q(12) )

C HC ACCELERATIONS, INERTIAL AXES - FT/S2
  EQUIVALENCE (DU( 1), DV1SN(1) )

C HC BODY ROLL, PITCH AND YAW ACCELERATIONS - RAD/SEC2
  EQUIVALENCE (DU( 4), DOM11(1) )

C LOAD BODY ROLL, PITCH AND YAW ACCELERATIONS - RAD/SEC2
  EQUIVALENCE (DU( 7), DOM22(1) )

C STRETCH COORDS, HOOK TO LOAD CG LINE SEGMENT COORDS
  EQUIVALENCE (DU(10), DDRA2S2(1) )

C HC BODY ROLL, PITCH AND YAW RATES - RAD/SEC
  EQUIVALENCE (U( 4), OM11(1) )

C LOAD BODY ROLL, PITCH AND YAW RATES - RAD/SEC
  EQUIVALENCE (U( 7), OM22(1) )

C FIRST DERIVATIVE, HOOK TO LOAD CG LINE SEGMENT
  EQUIVALENCE (U(10), DRA2S2(1) )

C HC EULER ROLL, PITCH AND YAW RATES - RAD/SEC
  EQUIVALENCE (DQ( 4), DPH1 )
  EQUIVALENCE (DQ( 5), DTH1 )
  EQUIVALENCE (DQ( 6), DPS1 )

```

```

C LOAD EULER ROLL, PITCH AND YAW RATES - RAD/SEC
    EQUIVALENCE (DQ( 7),  DPH2 )
    EQUIVALENCE (DQ( 8),  DTH2 )
    EQUIVALENCE (DQ( 9),  DPS2 )

C HC INERTIAL POSITION VECTOR
    EQUIVALENCE (Q( 1),  R1SN(1) )

C HC EULER ROLL, PITCH AND YAW ANGLES - RAD
    EQUIVALENCE (Q( 4),  PH1 )
    EQUIVALENCE (Q( 5),  TH1 )
C Note: PS1 included separately where needed
C LOAD EULER ROLL, PITCH AND YAW ANGLES - RAD
    EQUIVALENCE (Q( 7),  PH2 )
    EQUIVALENCE (Q( 8),  TH2 )
    EQUIVALENCE (Q( 9),  PS2 )

C HOOK TO LOAD CG LINE SEGMENT, LOAD BODY AXES
    EQUIVALENCE (Q(10),  RA2S2(1) )

C STRIKE variables used by: GHSLMC, GHSLMC_IC, CONEXAERO, NRT_UNC3_OUT
    COMMON /XFLOAT/ A(500)
    EQUIVALENCE (A( 1), PH1DEG )
    EQUIVALENCE (A( 2), TH1DEG )
    EQUIVALENCE (A( 3), PS1DEG )
    EQUIVALENCE (A( 76), WN(1) )
    EQUIVALENCE (A( 97), G )
    EQUIVALENCE (A(116), I1XX )
    EQUIVALENCE (A(117), I1YY )
    EQUIVALENCE (A(118), I1ZZ )
    EQUIVALENCE (A(119), I1XZ )
    EQUIVALENCE (A(168), DT )
    EQUIVALENCE (A(183), RHO )
    EQUIVALENCE (A(303), T )
    EQUIVALENCE (A(359), R2D )

C STRIKE Switches used by: GHSLMC, CONEXAERO
    COMMON /IFIXED/ IA(250)
    EQUIVALENCE (IA(1), IMODE )

C Variables for data output used by: NRT_UNC3_OUT
    COMMON /FCSCOM/ FCS(100)
    EQUIVALENCE(FCS(16), RSAS)
    EQUIVALENCE(FCS(17), PSAS)
    EQUIVALENCE(FCS(18), YSAS)
    EQUIVALENCE(FCS(41), DMIXA)
    EQUIVALENCE(FCS(42), DMIXB)
    EQUIVALENCE(FCS(43), DMIXC)
    EQUIVALENCE(FCS(44), DMIXP)
    EQUIVALENCE(FCS(49), PSFWD)
    EQUIVALENCE(FCS(50), PSAFT)
    EQUIVALENCE(FCS(51), PSLAT)
    EQUIVALENCE(FCS(52), PSTR)

```

C Variables for wake computations and locations used by: CONEXAERO,  
C NRT\_UNC3\_OUT

```
COMMON /ROCOM/ RC(440)
EQUIVALENCE (RC( 51), XMUXH )
EQUIVALENCE (RC( 52), XMUYH )
EQUIVALENCE (RC( 53), XMUZH )
EQUIVALENCE (RC( 54), XMUXS )
EQUIVALENCE (RC( 55), XMUYS )
EQUIVALENCE (RC( 56), XMUZS )
EQUIVALENCE (RC( 70), XLAMDA )
EQUIVALENCE (RC( 71), CTA )
EQUIVALENCE (RC( 80), A1F )
EQUIVALENCE (RC( 81), B1F )
EQUIVALENCE (RC(117), OMEGA )
EQUIVALENCE (RC(218), FSCG )
EQUIVALENCE (RC(219), WLCG )
EQUIVALENCE (RC(220), BLCG )
EQUIVALENCE (RC(221), RMR )
EQUIVALENCE (RC(223), W1 )
EQUIVALENCE (RC(280), WBLADE )
```

```
COMMON /IRCOM/ IRC(20)
EQUIVALENCE (IRC(10), NBS)
```

C Control History variables used by: NRT\_UNC3\_OUT, PILOT

```
COMMON /RCON/ RO(70)
EQUIVALENCE (RO(13), XA )
EQUIVALENCE (RO(14), XAAD )
EQUIVALENCE (RO(25), XB )
EQUIVALENCE (RO(26), XBAD )
EQUIVALENCE (RO(37), XP )
EQUIVALENCE (RO(38), XPAD )
EQUIVALENCE (RO( 1), XC )
EQUIVALENCE (RO( 2), XCAD )

COMMON /LFIXED/ LLOAD(15)
EQUIVALENCE (LLOAD( 1), STRETCH )
EQUIVALENCE (LLOAD( 2), NSTORE )
EQUIVALENCE (LLOAD( 3), NC )
EQUIVALENCE (LLOAD( 4), IAERSL )
EQUIVALENCE (LLOAD( 5), ILOAD )
EQUIVALENCE (LLOAD( 6), IPILOT )
EQUIVALENCE (LLOAD( 7), IWAKE )
EQUIVALENCE (LLOAD( 8), ISWIRL )
EQUIVALENCE (LLOAD( 9), IDATA )
EQUIVALENCE (LLOAD(10), QUAD )
EQUIVALENCE (LLOAD(11), AXIS )
```

C Load Description, Control History filename, and output filename

```
C used by: GHSL_INIT, NRT_UNC3_OUT, NRT_UNC3_IN
CHARACTER*40 LOADNAME, CHFILE, DATAFILE
COMMON /CSLLNCMN/ LOADNAME
COMMON /CSLCFCMN/ CHFILE
COMMON /CSLDFCMN/ DATAFILE
```

## APPENDIX C. DATA COLLECTION AND PROCESSING AUTOMATION

### C.1. DATA COLLECTION AUTOMATION

For each flight data point to be tested, Gen Hel/SL is run at the actual helicopter fuel weights reported on the post-flight debrief cards at the time of each frequency sweep. Typically, three simulation runs were used for each flight condition. Once each simulation runs is complete, the *datafile.dat* and *datafile.da2* output time histories are converted to UNC3 file format for use by CIFER<sup>®</sup> and XPLOT.

Although normally configured through the use of CIFER<sup>®</sup>'s internal user interface, FRESPID, MISOSA, and COMPOSITE COM files are created by editing templates files, substituting case specific information in for generic place-holders located in the templates. The COM files are data files used by CIFER<sup>®</sup> to direct the analysis, setting options such as the input and output channels, window sizes, and directory paths.

This method would not work for creating FRESPID cases for actual flight time history files, because the small differences in data rate for each file are detected and averaged by READMIS in CIFER<sup>®</sup> during case entry with the provided user interface. Simulation data files, however, provided data at consistent (0.01 seconds used here) intervals therefore do not require individual processed by READMIS; they are all exactly the same.

Once the COM files are complete, the script creates the file *batch\_master*, another C-Shell script that establishes a CIFER<sup>®</sup> environment in the computer and sequentially processes a list of COM files. In this way the operator can command a large number of Gen Hel/SL simulation runs followed by multiple FRESPID, MISOSA, and COMPOSITE analyses, effectively converting time history files into frequency response data and writing them to the CIFER<sup>®</sup> database for further processing.

The C-Shell scripts, named `scriptcase`, where *case* is the abbreviation for the load configuration (i.e. `nl`, `4kn`, `4cd`, etc.) also contain the instructions to launch the Expect scripts which interface with CIPHER® to produce the final results. Each script follows the same basic outline, but contains unique values of the `outfile`, `fuel`, `axis`, and `air` variables particular to each load configuration. The `outfile` is used to capture the screen output from Gen Hel/SL and `runghsl.dat` for troubleshooting and verification of the data collection run. The `fuel`, `axis`, and `air` variables set the fuel load, control input axis, and airspeed (in tens of knots) for the load configuration data points desired.

The files required by the data collection script to execute are:

- 1) `batch_master.old` – template file for the CIPHER batch control file
- 2) `ghsl.slmc` – template configuration data file for the single point, multi-cable load configuration (`ghsl.noload` for the no load condition)
- 3) `genhel` – FORTRAN compiled executable Gen Hel/SL program
- 4) `runghsl.dat` – FORTRAN compiled executable file from `ghsl_dat.f` source code
- 5) `fre_template` – template file for the FRESPID cases (`fre_template.nl` for the no load condition)
- 6) `mis_template` – template file for the MISOSA cases
- 7) `com_template` – template file for the COMPOSITE cases
- 8) Data processing scripts as required (discussed later)

A sample data collection automation script, in this case for the 4K Block with no load aerodynamics, is given below. Note that although C-Shell scripts do not have specific line lengths, continuing lines are shown indented three spaces for clarity. Each script follows essentially the same pattern, with small modifications to fit the specific case.

```
#!/bin/csh
# 'script4kn'
# Peter Tyson 19 JAN 99
# batch program for multiple GenHel runs, convert output files to "xp"
```

```

# format, transfer files to "tfdata" directory, make CIPHER "COM" files,
# set up "batch_master" and run CIPHER cases.
# this file for 4K BLOCK SLMC cases, lateral/longitudinal sweeps, no
# load aerodynamics

cd ~/cifer/jobs ; cp batch_master.old batch_master
mv *.COM.* oldjobs ; rm *.OUT.* ; cd ~/GenHel/batch

# run configuration settings
set outfile = (GH4KN.log)
set fuel = (1130 1080 1020 1260 1420 1220 2080 2010 1970 810 770 2190
1510 1480 1290 1820 1770 1730 1890 1850 1810 1280 1230 1200)
set air = (0 0 3 3 5 5 8 8)
set axis = (1 2 1 2 1 2 1 2)

# log file header
date +%D %t%T" > $outfile
echo "\nLog file created by script4kn\nfuel = ${fuel}\nair = ${air}\naxis = ${axis}\n\n" >> $outfile

foreach num (1 2 3 4 5 6 7 8)
foreach run (1 2 3)
@ fueln++
if ($axis[$num] == '1') set axisname = (A)
if ($axis[$num] == '2') set axisname = (B)
set fname = "GH${axisname}4KN${air[$num]}.srun"
cd ~/GenHel/batch

# insert case information into the hcddata and slrun namelists in the
# ghsl.dat configuration file
sed -e "s/FWT =/FWT = ${fuel[$fueln]}.0/" -e "s/AIRSPEED =/AIRSPEED = ${air[$num]}.0,/" -e "s/AXIS =/AXIS = ${axis[$num]}/" -e "s/DATAFILE =/DATAFILE = \"'${fname}'\"/" ghsl.slmc > ghsl.dat

# add the load configuration information in the sldata namelist
echo " &SLDATA\n LOADNAME = '4K STEEL BLOCK',\n W2 = 3895.0,\n I2XX = 103.0,\n I2YY = 103.0,\n I2ZZ = 174.0,\n I2XZ = 0.0,\n KS = 96 45.0,\n CS = 22.0, RA2PO2 = 0.0, 0.0, 16.4391,\n R2P2SO2 = 0.0, 0.0, 0.0,\n R2P2S2 = 0.0, 0.0, 0.0,\n NC = 4 R2PJ2 = 1.32, -1.32, -0.61,\n 1.32, 1.32, -0.61, -1.32, -1.32, -0.61,\n -1.32, 1.32, -0.61, 12*0,\n DOQ = 0.0,\n &END" >> ghsl.dat
echo -n " $fname" > fname.dat

# run Gen Hel/SL and convert output file to UNC3 format
nrt/genhel >>& $outfile ; runghsl.dat >>& $outfile ; rm $fname.da?
mv $fname.xp ~/cifer/jobs/tfdata ; mv $fname.out outfiles
end

# insert names of cases to be run into the batch_master file
cd ~/cifer/jobs ; set rname = "GH${axisname}4KN${air[$num]}"
echo
"FRE_$rname.COM.01\nMIS_$rname.COM.01\nCOM_$rname.COM.01\nCOM_${rname}D.COM.01" >> batch_master
echo "date +%D %t%T" >> batch_master

```



```

# create descriptive name for run for CIFER database entries
if ($air[$num] == '0') set spd = (HVR)
if ($air[$num] == '3') set spd = (30KT)
if ($air[$num] == '5') set spd = (50KT)
if ($air[$num] == '8') set spd = (80KT)
if ($axis[$num] == '1') set swp = (LAT)
if ($axis[$num] == '2') set swp = (LON)
set descr = (GH $swp 4K BLOCK, NO AERO, $spd)

# create FRESPID, MISOSA, and COMPOSITE ".COM." files, unique for each
# input axis
if ($axis[$num] == '1') then
    sed -e "s/CaseName/$rname/g" -e "s/DescriptHere/$descr/" -e "s/OUT1/
        P /" -e "s/OUT2/P2P /" -e "s/OUTA/pl /" -e "s/OUTB/p2p /" -e "
        s/AAAAAA/1 0 0 0 0 0/" -e "s/BBBBBB/0 0 0 0 1 0/" -e "s
        /CCCCC/0 0 0 0 0 0/" fre_template > FRE_$rname.COM.01
    sed -e "s/CaseName/$rname/g" -e "s/DescriptHere/$descr/" -e "s/CONT1
        /LAT/" -e "s/CONT2/LON/" -e "s/OUT1/P /" mis_template > MIS_$rnam
        e.COM.01
    sed -e "s/CaseName/$rname/g" -e "s/DescriptHere/$descr/" -e "s/CON1/
        LAT /" -e "s/OUT1/P /" com_template > COM_$rname.COM.01
    sed -e "s/CaseName/$rname/g" -e "s/DescriptHere/$descr/" -e "s/CON1/
        LTMI/" -e "s/CON2/LAT /" -e "s/OUT1/RSAS/" -e "s/OUT2/P2P /" comd
        _template > COM_${rname}D.COM.01
else if ($axis[$num] == '2') then
    sed -e "s/CaseName/$rname/g" -e "s/DescriptHere/$descr/" -e "s/OUT1/
        Q /" -e "s/OUT2/Q2P /" -e "s/OUTA/q1 /" -e "s/OUTB/q2p /" -e "
        s/AAAAAA/0 1 0 0 0 0/" -e "s/BBBBBB/0 0 0 0 0 0/" -e "s
        /CCCCC/0 0 0 0 0 1/" fre_template > FRE_$rname.COM.01
    sed -e "s/CaseName/$rname/g" -e "s/DescriptHere/$descr/" -e "s/CONT1
        /LON/" -e "s/CONT2/LAT/" -e "s/OUT1/Q /" mis_template > MIS_$rnam
        e.COM.01
    sed -e "s/CaseName/$rname/g" -e "s/DescriptHere/$descr/" -e "s/CON1/
        LON /" -e "s/OUT1/Q /" com_template > COM_$rname.COM.01
    sed -e "s/CaseName/$rname/g" -e "s/DescriptHere/$descr/" -e "s/CON1/
        LGMI/" -e "s/CON2/LON /" -e "s/OUT1/PSAS/" -e "s/OUT2/Q2P /" comd
        _template > COM_${rname}D.COM.01
endif
end
cd ~/cifer/jobs ; chmod +x *.COM.01
echo "\necho 'Complete with CIFER batch jobs'" >> batch_master

# execute batch_master script to run CIFER cases
echo "Running CIFER cases ...." ; batch_master >&! batch.log

# execute TCL/Expect scripts for data processing
cd ~/cifer/tcltkscripts
echo "Running HQ analysis ...." ; hqgh.4kn
echo "      SM analysis ...." ; smgh.4kn
echo "      LC analysis ...." ; lcgh.4kn
echo "Complete."
date +"%t%D %tT"

```

## **C.2. DATA PROCESSING AUTOMATION**

After the Gen Hel/SL runs and the CIPHER<sup>®</sup> frequency responses are generated and added to the CIPHER<sup>®</sup> database, a combination of C-Shell and Expect scripts are used to interact with CIPHER<sup>®</sup> and extract performance parameters for each case. These analyses are divided into the three phases, HQ, SM and LC (load characteristics).

### **1. Handling Qualities Analysis**

The sample Handling Qualities analysis script `hqgh.4kn` shown below uses variables set by the user to control the operation of the Expect script through environment variables set in the file `hqset.tcl`. As with the data collection script, the variables determine the aircraft (GH for simulation or FT for flight data), load (3 letter abbreviation), axis, and airspeed (in tens of knots) for each analysis case (refer to Appendix G for case naming conventions and CIPHER<sup>®</sup> database contents). Other variables are used to direct the Handling Qualities determination, performed by the Expect script using CIPHER<sup>®</sup>'s text-based user interface. The HQ frequency response is given a gain and time delay correction as desired, and the analysis can cycle through the different correction factors (no correction, airspeed specific, or average value) as desired.

The output from the script is directed to a text file, which for the sample script is named `hqGH4KN.out`. If desired the HQ frequency response plots generated by CIPHER (the magnitude, phase and coherence plot and the least squares fit plot) can be saved as Postscript files. Finally, the HQ frequency response can be saved as an ASCII text file for further processing, such as use in MATLAB.

The Expect script for the HQ analysis is also given below. It was modified from the Real-Time CIPHER<sup>®</sup> Graphical User Interface being developed at NASA Ames by Miss Ranjana Sahai for use in processing flight time histories during data collection for immediate system parameter identification.

```

#!/bin/csh
# 'hqgh.4kn' CIPHER Handling Qualities Extractor
# Peter Tyson 12 Feb 99, mod 5 Mar 99

set acft = (GH) ; set load = (4KN)
set hqpath = (~/cifer/tcltkscripts)
set plotpath = (~/cifer/jobs/plots) ; rm $plotpath/BAN*
set hqfile = $hqpath/hq$acft$load.log

# configuration parameters
set axis = (1 1 1 1 2 2 2 2)
set air = (0 3 5 8 0 3 5 8)
set shift = (0 -360 0 0 0 0 0 0)
echo "Handling Qualities Parameters, Gen Hel / SL Simulation, Load: ${load}" > $hqfile ; date +%D %t%T >> $hqfile

foreach run (1 2 3)

# incorporate Gen Hel/SL correction factors
if ($run == '1') then
    set gain = (1 1 1 1 1 1 1 1)
    set delay = (0 0 0 0 0 0 0 0)
    set name = (_HQ) ; set maxfreq = (20.0)
    echo "No Gain and Time Delay Correction\n" >> $hqfile
else if ($run == '2') then
    set gain = (0.929 1.040 1.030 1.010 .767 .762 .854 .853)
    set delay = (.0427 .0480 .0484 .0549 .0546 .0336 .0470 .0688)
    set name = (_HQAS) ; set maxfreq = (15.0)
    echo "Airspeed Specific Gain and Time Delay Correction\n" >> $hqfile
else
    set gain = (1.000 1.000 1.000 1.000 .809 .809 .809 .809)
    set delay = (.0485 .0485 .0485 .0485 .0510 .0510 .0510 .0510)
    set name = (_HQAVG) ; set maxfreq = (15.0)
    echo "Average Gain and Time Delay Correction\n" >> $hqfile
endif

foreach num (1 2 3 4 5 6 7 8)
echo "set hqname ${name}\nset hqgain ${gain[$num]}\nset hqshift ${shift[$num]}\nset hqdelay ${delay[$num]}" > $hqpath/hqset.tcl
echo "set hqsave 1\nset hqminon 0\nset hqmin 0.5\nset hqmaxon 0\nset hqmax ${maxfreq}" >> $hqpath/hqset.tcl

if ($axis[$num] == '1') then
    echo "set lower 6\nset upper 10\nset case ${acft}A${load}${air[$num]}\nset tail _COM_ABCDE_LAT_P" >> $hqpath/hqset.tcl
    set case = (${acft}A${load}${air[$num]})
else
    echo "set lower 4\nset upper 6\nset case ${acft}B${load}${air[$num]}\nset tail _COM_ABCDE_LON_Q" >> $hqpath/hqset.tcl
    set case = (${acft}B${load}${air[$num]})
endif

hq.exp > out.log ; rm out.log
sed -n -e 1p -e 4,5p -e 7,8p -e 12,14p $hqpath/hqout >> $hqfile

```

```
mv $plotpath/BAN_${case}_MPC.PSC.01 $plotpath/hq${case}.${run}1
mv $plotpath/BAN_${case}_PHD.PSC.01 $plotpath/hq${case}.${run}2
```

```
end
end
```

```
#!/usr/bin/expect
# hq.exp
# based on Real Time CIFER GUI by Ranjana Sahai, NASA Ames
# modified by Peter Tyson 12 Feb 99
#
set hqpath {~/cifer/tcltkscripts}
source $hqpath/hqset.tcl
#
spawn cifer
expect "Enter <cr> for menu." ; exp_send "\r"
expect "Input:" ; exp_send "8\r"
expect "Enter*end]:" ; exp_send "D\r"
expect "Input:" ; exp_send "$case$tail\r"
expect "denominator?*:" ; exp_send "Y\r"
expect "to the denominator*:" ; exp_send -- "-1\r"
expect "a gain correction?" ; exp_send "Y\r"
expect "gain correction :" ; exp_send "$hqgain\r"
expect "a phase shift?*:" ; exp_send "Y\r"
expect "phase shift*degrees):" ; exp_send -- "$hqshift\r"
expect "a time delay?*:" ; exp_send "Y\r"
expect "delay*seconds):" ; exp_send -- "$hqdelay\r"
expect "to skip):" ; exp_send "0.5\r"
expect "to skip):"
set HQfileId [open $hqpath/hqout w+ 0666]
puts $HQfileId "Casename: $case$tail * s^-1"
puts $HQfileId "$expect_out(buffer)"
close $HQfileId
exp_send -- "-1\r"
expect "to skip):" ; exp_send -- "-1\r"
expect "and coherence plots?:" ; exp_send "Y\r"
expect "for default):"
if {$hqminon == 1} {
    exp_send -- "$hqmin\r"
} elseif {$hqminon == 0} {
    exp_send "\r"
}
expect "for default):"
if {$hqmaxon == 1} {
    exp_send -- "$hqmax\r"
} elseif {$hqmaxon == 0} {
    exp_send "\r"
}
expect "*" ; exp_send "\r"
expect "or C(omprs):" ; exp_send "P\r"
expect "crossover values? :" ; exp_send "N\r"
expect "analysis)?:" ; exp_send "Y\r"
```

```

expect "for default):"
if {$hqminon == 1} {
    exp_send -- "$hqmin\r"
} elseif {$hqminon == 0} {
    exp_send "\r"
}
expect "for default):"
if {$hqmaxon == 1} {
    exp_send -- "$hqmax\r"
} elseif {$hqmaxon == 0} {
    exp_send "12\r"
}
expect "*" ; exp_send "\r"
expect "or C(omprs):" ; exp_send "\r"
expect "HQ analysis):" ; exp_send "Y\r"
expect "if done.):" ; exp_send -- "$lower\r"
expect "for default):" ; exp_send -- "$upper\r"
expect "coherence weighting):" ; exp_send "Y\r"
expect "*" ; exp_send "\r"
expect "or C(omprs):" ; exp_send "P\r"
expect "HQ analysis):" ; exp_send "N\r"
expect "frequency response*:"
if {$hqsave == 1} {
    exp_send "Y\r"
    expect "Input:"
    exp_send -- "$case$hqname\r"
} elseif {$hqsave == 0} {
    exp_send "N\r"
}
expect "to end):" ; exp_send "\r"
expect "to continue" ; exp_send "\r"
expect "Input:"
if {$hqsave == 1} {
    exp_send "20\r"
    expect "name:" ; exp_send -- "$case$hqname\r"
    expect "(or <CR>):" ; exp_send "\r"
    expect "F(ile)*:" ; exp_send "\r"
    expect "Input:" ; exp_send "\r"
    expect "name:" ; exp_send "\r"
    expect "Input:" ; exp_send "\r"
} elseif {$hqsave == 0} {
    exp_send "\r"
}
#
# end hq.exp

```

## 2. Stability Margin Analysis

Similar to the HQ analysis, the Stability Margin analysis is done through two scripts, *smft.case* or *smgh.case* and *sm.exp*. In the C-Shell script the same variables as with data collection and HQ analysis are used to set up the configuration, and each run is considered in order through the use of the *foreach* command. The example script shown below is for the case GH4KN, followed by the Expect script *sm.exp*.

```
#!/bin/csh
# 'smgh.4kn' CIFER Stability Margins Extractor
# Peter Tyson 12 Feb 99

set acft = (GH) ; set load = (4KN)
set smpath = (~/.cifer/tcltkscripts)
set plotpath = (~/.cifer/jobs/plots) ; rm $plotpath/BAN*

# configuration variables
set axis = (1 1 1 1 2 2 2 2)
set air = (0 3 5 8 0 3 5 8)
set shift = (-180 -180 180 -180 -180 180 -180 -180) # shift for 4KN
echo "Stability Margin Parameters\nGen Hel / Slung Load Simulation,
Load: ${load}\nShift = $shift\n" > $smpath/sm$acft$load.out
date +"%D %t%T" >> $smpath/sm$acft$load.out

foreach run (1 2 3)
if ($run == '1') then
    set gain = (1 1 1 1 1 1 1 1)
    set delay = (0 0 0 0 0 0 0 0)
    echo "No Gain and Time Delay Corr\n" >> $smpath/sm$acft$load.out
    echo "set smmaxfreq 20\nset smname _SM" > $smpath/smset.tcl
else if ($run == '2') then
    set gain = (0.929 1.040 1.030 1.010 0.767 0.762 0.854 0.853)
    set delay = (.0427 .0480 .0484 .0549 .0546 .0336 .0470 .0688)
    echo "Airspeed Specific Gain and Time Delay Corr\n" >> $smpath/sm$acft$load.out
    echo "set smmaxfreq 15\nset smname _SMAS" > $smpath/smset.tcl
else
    set gain = (1.000 1.000 1.000 1.000 0.809 0.809 0.809 0.809)
    set delay = (.0485 .0485 .0485 .0485 .0510 .0510 .0510 .0510)
    echo "Average Gain and Time Delay Correction\n" >> $smpath/sm$acft$load.out
    echo "set smmaxfreq 15\nset smname _SMAVG" > $smpath/smset.tcl
endif

foreach num (1 2 3 4 5 6 7 8)
echo "set smsave 1\nset smgain ${gain[$num]}\nset smshift ${shift[$num]}
\nset smdelay ${delay[$num]}\nset smfreqon 0\nset smminfreq 1.0" >>
    $smpath/smset.tcl
```

```

if ($axis[$num] == '1') then
    echo "set case ${acft}A${load}${air[$num]}\nset tail _COM_ABCDE_LTMI
    _RSAS" >> $smpath/smset.tcl
    set case = (${acft}A${load}${air[$num]})
else
    echo "set case ${acft}B${load}${air[$num]}\nset tail _COM_ABCDE_LGMI
    _PSAS" >> $smpath/smset.tcl
    set case = (${acft}B${load}${air[$num]})
endif

```

```

sm.exp > out.log ; rm out.log
sed -e /'Enter'/d -e /'Start'/d -e /'End'/d -e /'^'...0\.'/d -e /'Number
    '/d -e /'Do you'/d -e /'crossover'/d -e /'crossings'/d $smpath/smout
    >> $smpath/sm$acft$load.out

```

```

mv $plotpath/BAN_${case}_MPC.PSC.01 $plotpath/sm${case}.${run}
end
end

```

```

#!/usr/bin/expect
# sm.exp
# based on Real Time CIFER GUI by Ranjana Sahai, NASA Ames
# modified by Peter Tyson 12 Feb 99
#
set smpath {~/cifer/tcltkscripts}
set plotpath {/d10/phtyson/plots}
source $smpath/smset.tcl
#
spawn cifer
expect "Enter <cr> for menu." ; exp_send "\r"
expect "Input:" ; exp_send "8\r"
expect "Enter*end]:" ; exp_send "D\r"
expect "Input:" ; exp_send "$case$tail\r"
expect "denominator?*"
set SMfileId [open $smpath/smout w+ 0666]
puts $SMfileId "$expect_out(buffer)"
exp_send "N\r"
expect "a gain correction?" ; exp_send "Y\r"
expect "gain correction :" ; exp_send "$smgain\r"
expect "a phase shift?*" ; exp_send "Y\r"
expect "phase shift*degrees):" ; exp_send -- "$smshift\r"
expect "a time delay?*" ; exp_send "Y\r"
expect "delay*seconds):" ; exp_send -- "$smdelay\r"
expect "to skip):" ; exp_send -- "-1\r"
expect "to skip):"
if {$smfreqon == 1} {
    exp_send "$smminfreq,$smmaxfreq\r"
} elseif {$smfreqon == 0} {
    exp_send "\r"
}
while 1 {
    expect {
        "No 0dB crossing found" break
    }
}

```

```

        "Phase margin*deg"
    }
    puts $SMfileId "$expect_out(buffer)"
    exp_send "\r"
}
while 1 {
    expect {
        "User can define new search range now" break
        "Gain Margin*dB"
    }
    puts $SMfileId "$expect_out(buffer)"
    exp_send "\r"
}
close $SMfileId
exp_send -- "-1\r"
expect "and coherence plots?:" ; exp_send "Y\r"
expect "for default):"
if {$smfreqon == 1} {
    exp_send -- "$smminfreq\r"
} elseif {$smfreqon == 0} {
    exp_send "\r"
}
expect "for default):"
if {$smfreqon == 1} {
    exp_send -- "$smmaxfreq\r"
} elseif {$smfreqon == 0} {
    exp_send "\r"
}
expect "*" ; exp_send "\r"
expect "or C(omprs):" ; exp_send "P\r"
expect "crossover values? :" ; exp_send "N\r"
expect "save this frequency response?*"
if {$smsave == 1} {
    exp_send "Y\r"
    expect "Input:"
    exp_send -- "$case$smname\r"
} elseif {$smsave == 0} {
    exp_send "N\r"
}
expect "to end]:" ; exp_send "\r"
expect "to continue..." ; exp_send "\r"
expect "Input:"
if {$smsave == 1} {
    exp_send "20\r"
    expect "name:" ; exp_send -- "$case$smname\r"
    expect "(or <CR>):" ; exp_send "\r"
    expect "F(ile)*:" ; exp_send "\r"
    expect "Input:" ; exp_send "\r"
    expect "name:" ; exp_send "\r"
    expect "Input:" ; exp_send "\r"
} elseif {$smsave == 0} {
    exp_send "\r"
}
}
# end sm.exp

```



### 3. Load Characteristics Analysis

Load Characteristics analysis is performed through two scripts, *lcft.case* or *lcgh.case* and *lc.exp*, samples of which are given below. The *shift* variable, always a multiple of  $360^\circ$ , is required to adjust for the phase straightening automatically done by CIPHER®, and has to be determined through trial and error. If the shift is not correct, the resulting 2<sup>nd</sup> order fit will have 2 real roots in the denominator instead of the imaginary pair from which damping ratio and natural frequency are extracted.

```
#!/bin/csh
# 'lcgh.4kn' CIPHER Load Characteristics Extractor
# Peter Tyson 12 Feb 99, mod 5 Mar 99

set acft = (GH) ; set load = (4KN)
set lcpath = (~/.cifer/tcltkscripts) ; set plotpath =
(~/.cifer/jobs/plots)
set axis = (1 1 1 2 2 2 2)
set air = (0 3 5 8 0 3 5 8)
set shift = (0 0 360 360 0 0 0 0)
echo "Load Characteristics, Gen Hel / Slung Load Simulation, Load: ${load}" > $lcpath/lc$acft$load.out
date +%D %t%T" >> $lcpath/lc$acft$load.out

foreach num (1 2 3 4 5 6 7 8)

echo "set lcsave 1\nset lcname _LC\nset lcgain 1\nset lcshift ${shift[$num]}\nset lcdelay 0" > $lcpath/lcset.tcl
echo "set lcfreqson 1\nset lcfreqs 0.5,2.5,50\nset lcdelayon Y\nset lcnegcoef Y" >> $lcpath/lcset.tcl
if ($axis[$num] == '1') then
    echo "set case ${acft}A${load}${air[$num]}\nset tail _COM_ABCDE_LAT_P2P" >> $lcpath/lcset.tcl
    set case = (${acft}A${load}${air[$num]})
else
    echo "set case ${acft}B${load}${air[$num]}\nset tail _COM_ABCDE_LON_Q2P" >> $lcpath/lcset.tcl
    set case = (${acft}B${load}${air[$num]})
endif

lc.exp > out.log ; rm out.log
sed -n -e 1,2p -e 4,5p -e 11,15p $lcpath/lcout >>
$lcpath/lc$acft$load.out

mv $plotpath/NAV_NAVFIT.PSC.01 $plotpath/lc$case.01
mv $plotpath/NAV_NAVFIT.PSC.02 $plotpath/lc$case.02
```

end

```
#!/usr/bin/expect
# lc.exp
# based on Real Time CIFER GUI by Ranjana Sahai, NASA Ames
# modified by Peter Tyson 12 Feb 99
#
set lcpath {~/cifer/tcltkscripts}
source $lcpath/lcset.tcl
#
spawn cifer
expect "Enter <cr> for menu."      ; exp_send "\r"
expect "Input:"                    ; exp_send "6\r"
expect "and brief prompts:"        ; exp_send "1\r"
expect "Input:"                    ; exp_send "1\r"
expect "Input:"                    ; exp_send "1\r"
expect "Enter response name:"      ; exp_send "$case$tail\r"
expect "weighting in fit*:"        ; exp_send "Y\r"
if {$lcfreqson == 1} {
    exp_send -- "$lcfreqs\r"
} elseif {$lcfreqson == 0} {
    exp_send "\r"
}
expect "gain multiplier*:"         ; exp_send "$lcgain\r"
expect "or denominator*:"         ; exp_send "N\r"
expect "a phase shift?*:"         ; exp_send "Y\r"
expect "phase shift*degrees):"    ; exp_send -- "$lcshift\r"
expect "low order system:"        ; exp_send "0,2\r"
expect "a time delay:"            ; exp_send "$lcdelay\r"
expect "time delay be free*:"     ; exp_send "$lcdelayon\r"
expect "negative coefficients*:"  ; exp_send "$lcnegcoef\r"
expect "Input:"                   ; exp_send "\r"
expect "Input:"                   ; exp_send "2\r"
expect "tabulated results*:"
set LCfileId [open $lcpath/lcout w+ 0666]
puts $LCfileId "Casename: $case$tail\n"
puts $LCfileId "$expect_out(buffer)"
close $LCfileId
exp_send "N\r"
expect "not to store : "          ; exp_send "\r"
expect "Generate Bode plots*:"   ; exp_send "Y\r"
expect "P*alaris):"              ; exp_send "P\r"
expect "P*alaris):"              ; exp_send "\r"
expect "Input:"                  ; exp_send "5\r"
expect "to continue..."         ; exp_send "\r"
expect "Input:"
if {$lcsave == 1} {
    exp_send "20\r"
    expect "name:"                ; exp_send -- "$case$tail\r"
    expect "(or <CR>):"            ; exp_send "\r"
    expect "F(file)*:"            ; exp_send "\r"
    expect "Input:"               ; exp_send -- "$case$lcname\r"
    expect "name:"                ; exp_send "\r"
}
```

```
        expect "Input:"                ; exp_send "\r"
    } elseif {$lcsave == 0} {
        exp_send "\r"
    }
    #
    # end lc.exp
```

## APPENDIX D. NO EXTERNAL LOAD DATA

This Appendix contains a summary of the data found for the no load condition from flight data as compared to Gen Hel. In Table D.1, the cases presented at each airspeed for the lateral and longitudinal axes are given as:

<b>Fi</b>	Flight, indirect method for computing Stability Margins
<b>Fd</b>	Flight, direct method for computing Stability Margins
<b>N</b>	Gen Hel, no gain or time delay correction factor
<b>S</b>	Gen Hel, airspeed-specific gain and time delay correction
<b>A</b>	Gen Hel, average gain and time delay correction factor

The frequency responses shown in Figures D.1 through D.8 are for the Flight and for Gen Hel with the average gain and time correction factor applied, except for Figure D.1(a) and Figure D.3(a), which show the uncorrected Gen Hel frequency responses for the hover condition in the lateral and longitudinal axes, respectively.

Figures D.1 through D.4 show the helicopter attitude response to aircraft attitude as required by the Handling Qualities analysis. Figures D.1 and D.3 present the overall frequency response across the range  $[0.5, 20]$  rad/sec and include magnitude and phase error functions with the proposed Level D criteria. Figures D.2 and D.4 give the same responses over the range  $[1, 10]$  rad/sec, providing a detailed view of the HQ parameter determination along with data coherence.

Figures D.5 through D.8 show the broken loop response used in the determination of Stability Margins from Flight, measured by both direct and indirect methods, and from the corrected Gen Hel simulation results. Figures D.5 and D.7 provide the data for the range  $[0.5, 15]$  rad/sec, and include the error functions. Figures D.6 and D.8 provide details on the SM determination.

Case			-180° Crossing (rad/s)	-180° Gain (dB)	-135° Band- width (rad/s)	6 dB Band- width (rad/s)	Phase Delay (s)	LS Fit Range (rad/s)	Gain Margin (dB)	Gain Crossing (rad/s)	Phase Margin (deg)	Phase Crossing (rad/s)
L A T E R A L	0	Fi	7.690	-32.57	5.218	4.278	0.1483	4-8	18.30	9.650	126.1 141.5	0.3407 0.9793
		Fd							12.15	10.12	123.3 136.9	0.3742 0.7400
		N	9.467	-35.94	6.067	6.239	0.1095		-	-	114.5 130.8	0.4212 2.198
		S	7.872	-33.58	5.278	4.379	0.1308		14.13	12.40	114.0 131.3	0.4260 1.717
		A	7.725	-32.60	5.060	3.875	0.1337		12.93	11.81	113.3 124.7	0.4212 2.198
	30	Fi	7.506	-32.59	4.184	4.380	0.1186	4-8	15.82	10.38	167.6 141.9	0.3936 0.8699
		Fd							12.29	10.90	165.3 139.8	0.4262 0.8124
		N	8.190	-35.54	7.109	5.398	0.1174		12.09	12.31	-	-
		S	7.018	-40.17	4.807	6.568	0.1414		7.683	8.599	-	-
		A	7.018	-40.51	4.803	6.567	0.1417		8.024	8.595	-	-
	50	Fi	7.543	-32.82	4.110	4.296	0.1142	4-8	14.93	9.692	155.8 135.1	0.2837 1.059
		Fd							11.43	10.44	154.3 132.7	0.2918 0.9767
		N	9.160	-35.25	5.822	5.629	0.1046		22.96	19.88	142.6 120.6	0.3918 2.301
		S	7.596	-32.01	4.745	3.718	0.1288		11.23	11.05	142.0 112.7	0.3894 2.419
		A	7.593	-32.26	4.743	3.715	0.1289		11.48	11.05	141.5 114.2	0.3918 2.301
	80	Fi	7.788	-32.78	4.133	4.303	0.1368	6-10	13.02	9.693	127.1 99.95	0.1871 1.854
		Fd							11.33	10.24	128.3 97.23	0.1920 1.899
		N	10.20	-36.24	5.841	5.958	0.0934		14.95	15.42	155.7 111.8	0.2449 2.581
		S	7.831	-32.97	4.470	4.187	0.1209		10.02	11.21	155.0 102.7	0.2439 2.648
		A	8.114	-33.40	4.557	4.372	0.1177		10.57	11.61	155.0 104.6	0.2449 2.581

Table D.1. Helicopter Response Summary: No Load Configuration, Lateral Axis

Case			-180° Cross-ing (rad/s)	-180° Gain (dB)	-135° Band- width (rad/s)	6 dB Band- width (rad/s)	Phase Delay (s)	LS Fit Range (rad/s)	Gain Margin (dB)	Gain Cross-ing (rad/s)	Phase Margin (deg)	Phase Cross-ing (rad/s)
LONGITUDINAL	0	Fi	4.291	-34.65	2.245	2.725	0.1895	2-6	21.74	6.501	48.85 86.93	0.2678 1.969
		Fd							15.30	8.068	41.58 73.45	0.2441 2.204
		N	5.244	-36.42	2.845	3.557	0.1626		18.44	11.33	98.31 77.66	0.2325 2.307
		S	4.534	-36.28	2.541	2.944	0.1899		15.82	7.640	100.2 79.30	0.2550 1.798
		A	4.568	-36.06	2.547	2.999	0.1881		15.63	7.787	99.82 74.21	0.2503 1.933
	30	Fi	4.534	-36.73	2.364	3.156	0.1730	2-6	18.78	6.371	160.4 106.7	0.4041 1.629
		Fd							13.59	8.435	152.1 95.82	0.4343 1.707
		N	4.440	-32.76	2.714	2.495	0.1628		12.56	8.932	119.1 86.83	0.6619 1.504
		S	4.384	-34.50	2.484	2.311	0.1796		8.704	5.936	120.8 98.90	0.7819 1.264
		A	4.354	-33.59	2.383	2.234	0.1883		7.862	5.838	120.3 93.98	0.7358 1.313
	50	Fi	4.237	-35.34	2.166	2.760	0.1588	2-6	22.73	7.153	150.7 107.3	0.3183 1.290
		Fd							13.42	8.508	143.1 89.99	0.3304 1.637
		N	5.137	-36.73	2.942	3.501	0.1662		18.60	11.90	114.9 86.41	0.8178 2.129
		S	4.466	-35.46	2.589	2.779	0.1897		16.27	8.225	109.8 90.62	0.9469 1.678
		A	4.444	-35.91	2.569	2.772	0.1917		16.46	8.091	107.7 95.15	1.011 1.511
	80	Fi	4.077	-33.89	0.440909	2.514	0.1585	2-6	25.61	5.400	-	-
		Fd							17.03	7.669	-	-
		N	6.881	-38.08	4.209	4.637	0.1379		14.92	12.24	95.01 85.82	0.2758 2.953
		S	5.418	-35.26	3.295	2.738	0.1723		11.31	7.744	94.98 89.79	0.3113 2.276
		A	5.828	-37.06	3.531	3.252	0.1634		12.86	8.525	95.62 94.79	0.3242 2.136

Table D.1. Continued, Longitudinal Axis

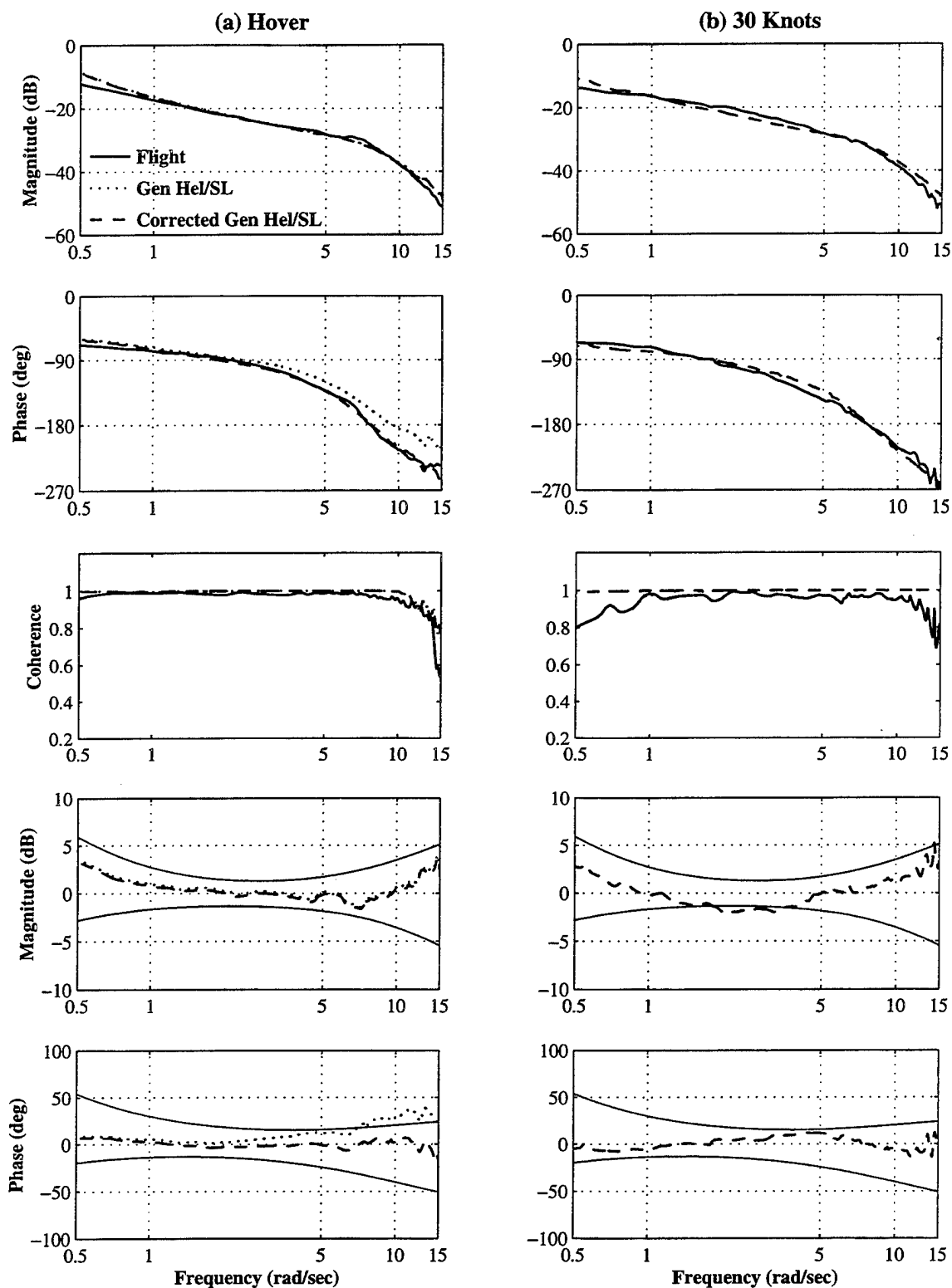


Figure D.1. No Load Handling Quality, Lateral Axis, (a) Hover, (b) 30 Knots

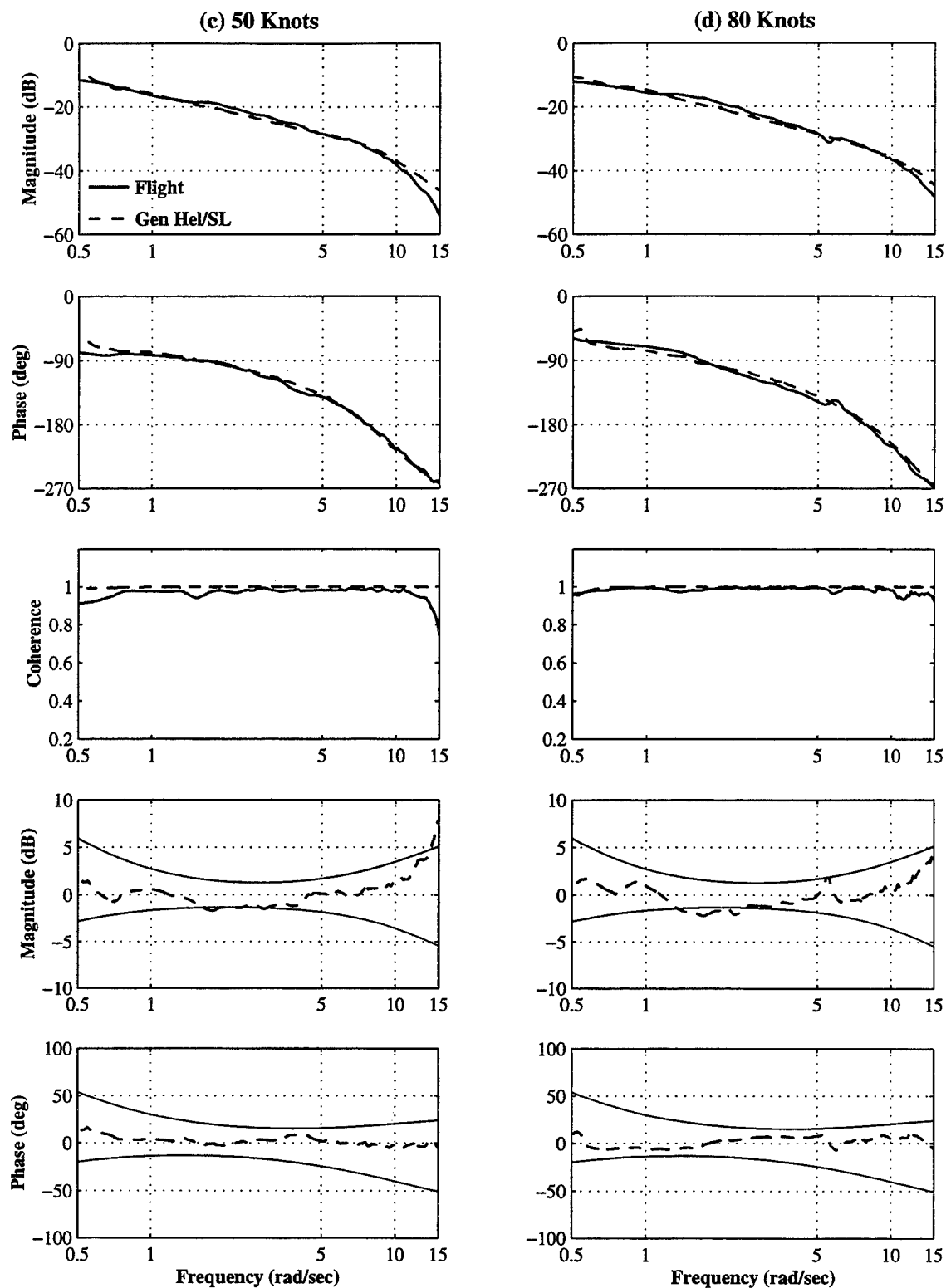


Figure D.1. Continued, (c) 50 Knots, (d) 80 Knots



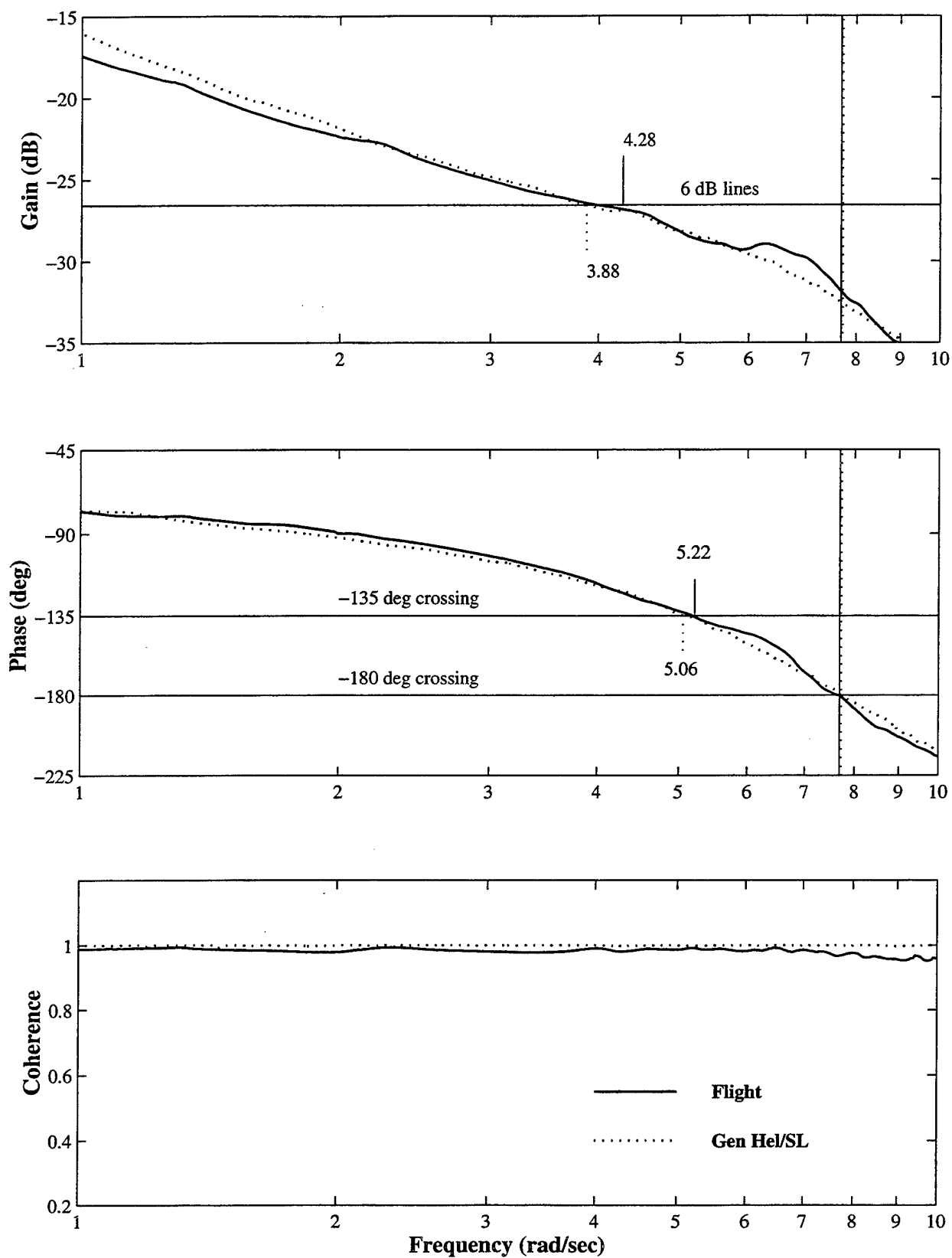


Figure D.2. No Load Handling Quality Determination, Lateral Axis, (a) Hover

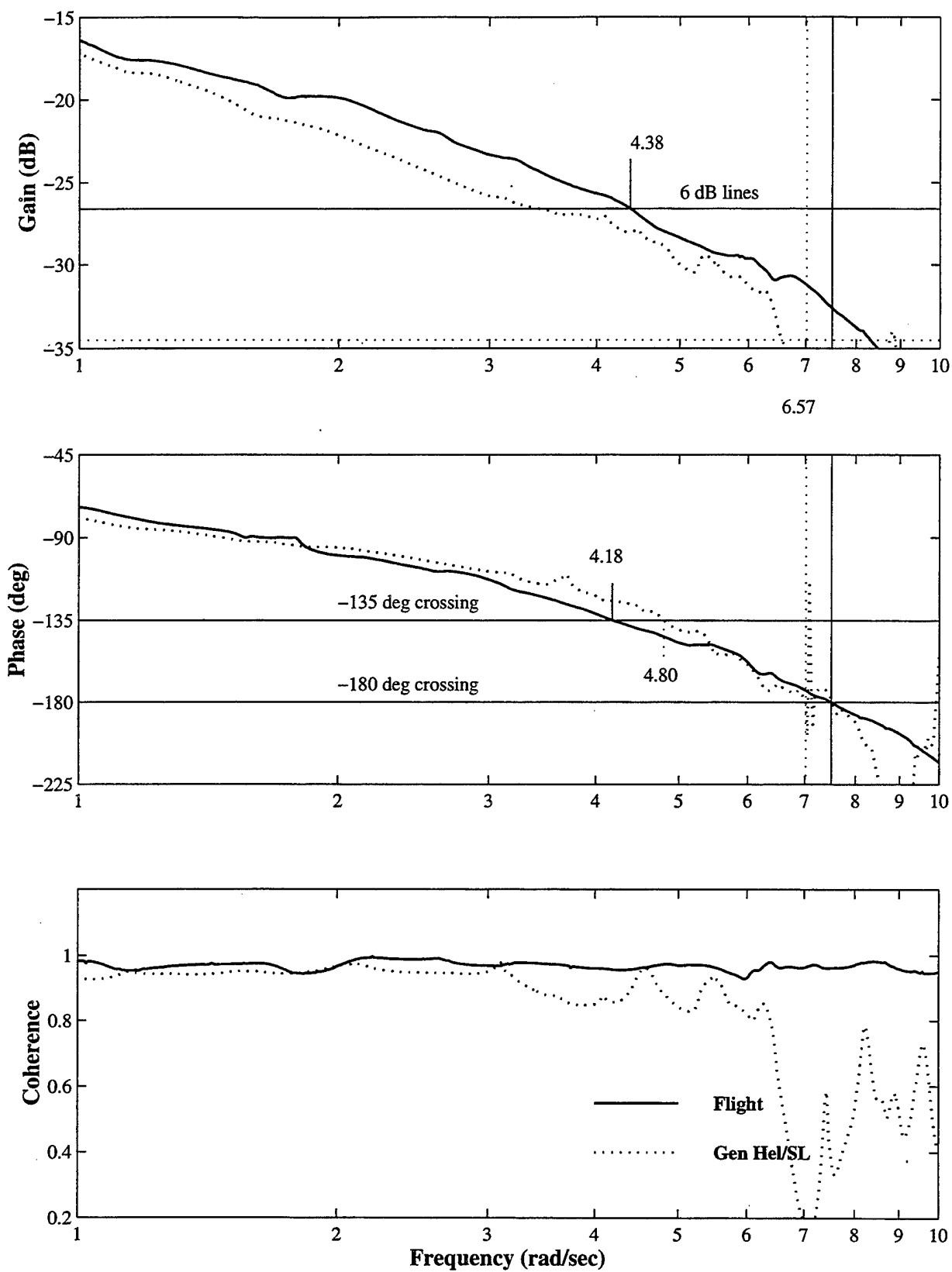


Figure D.2. Continued, (b) 30 Knots

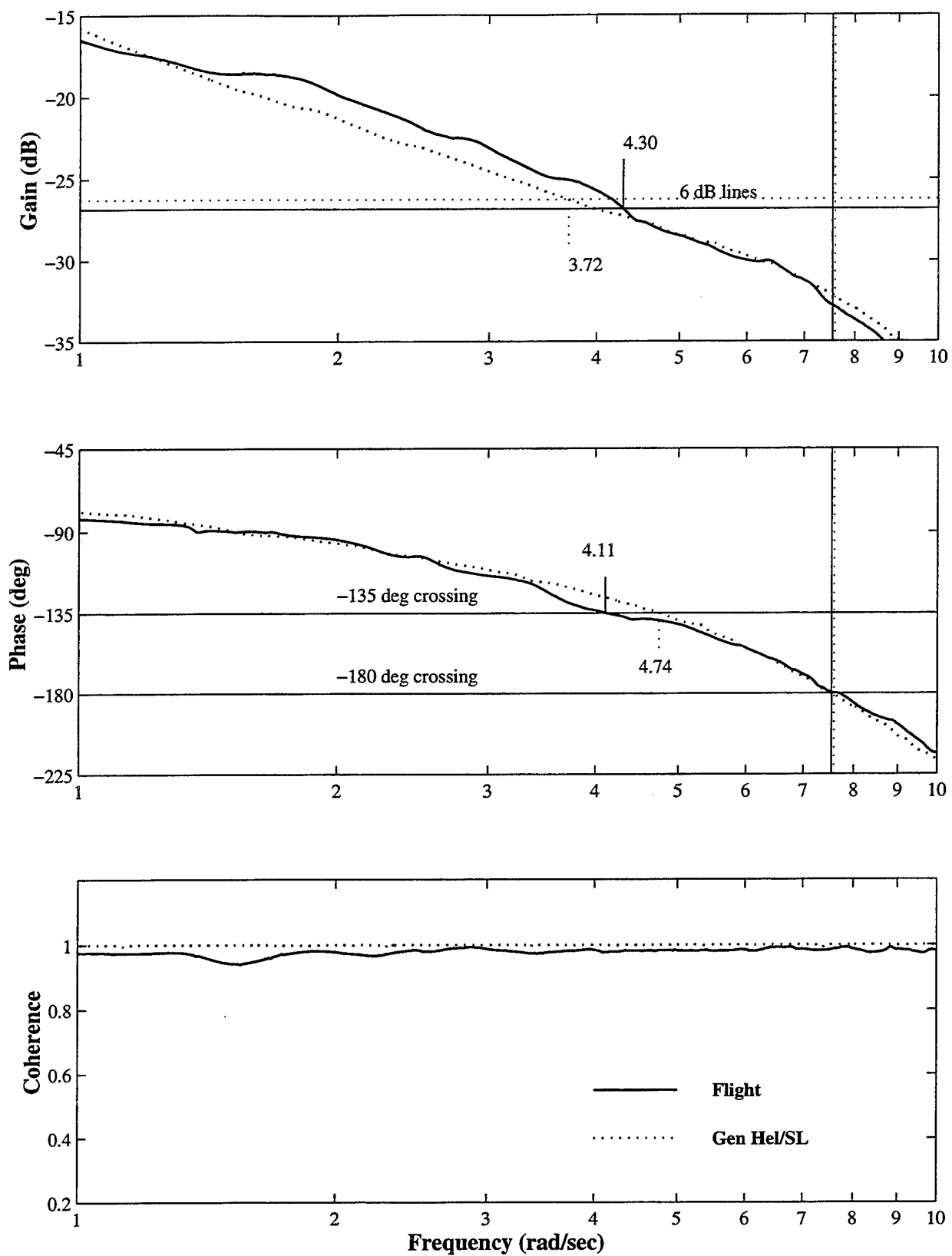


Figure D.2. Continued, (c) 50 Knots

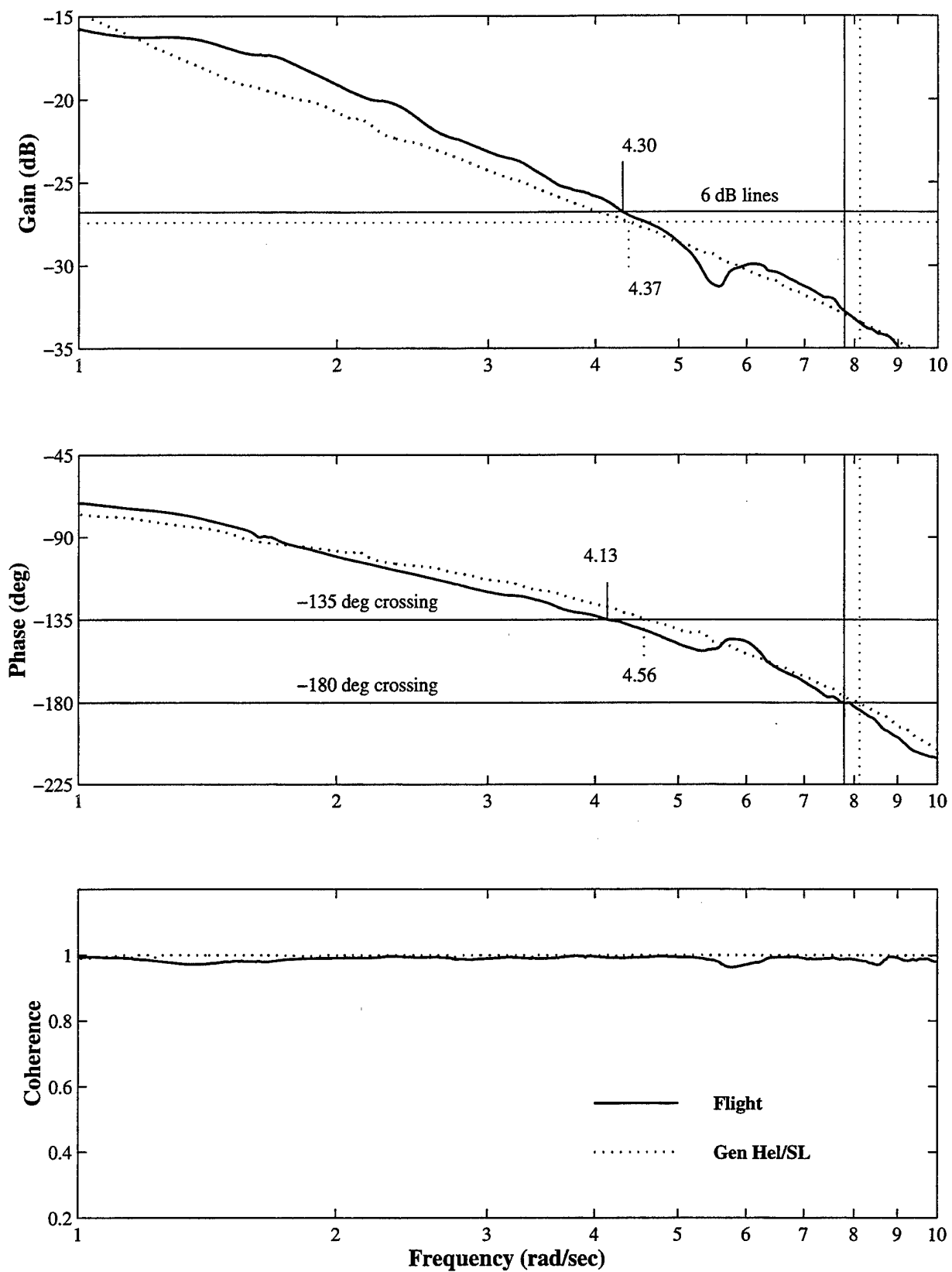


Figure D.2. Continued, (d) 80 Knots

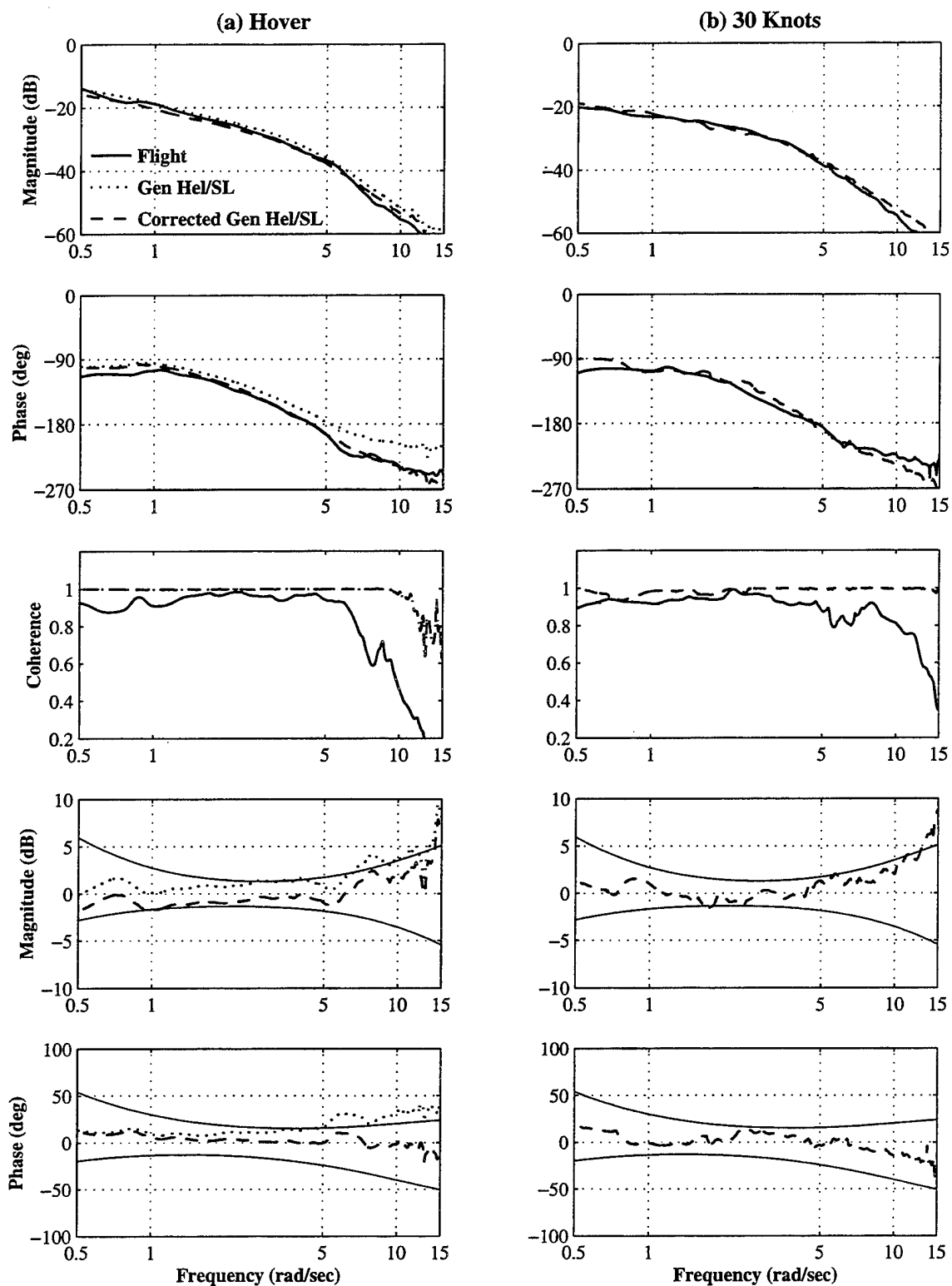


Figure D.3. No Load Handling Quality, Longitudinal Axis, (a) Hover, (b) 30 Knots

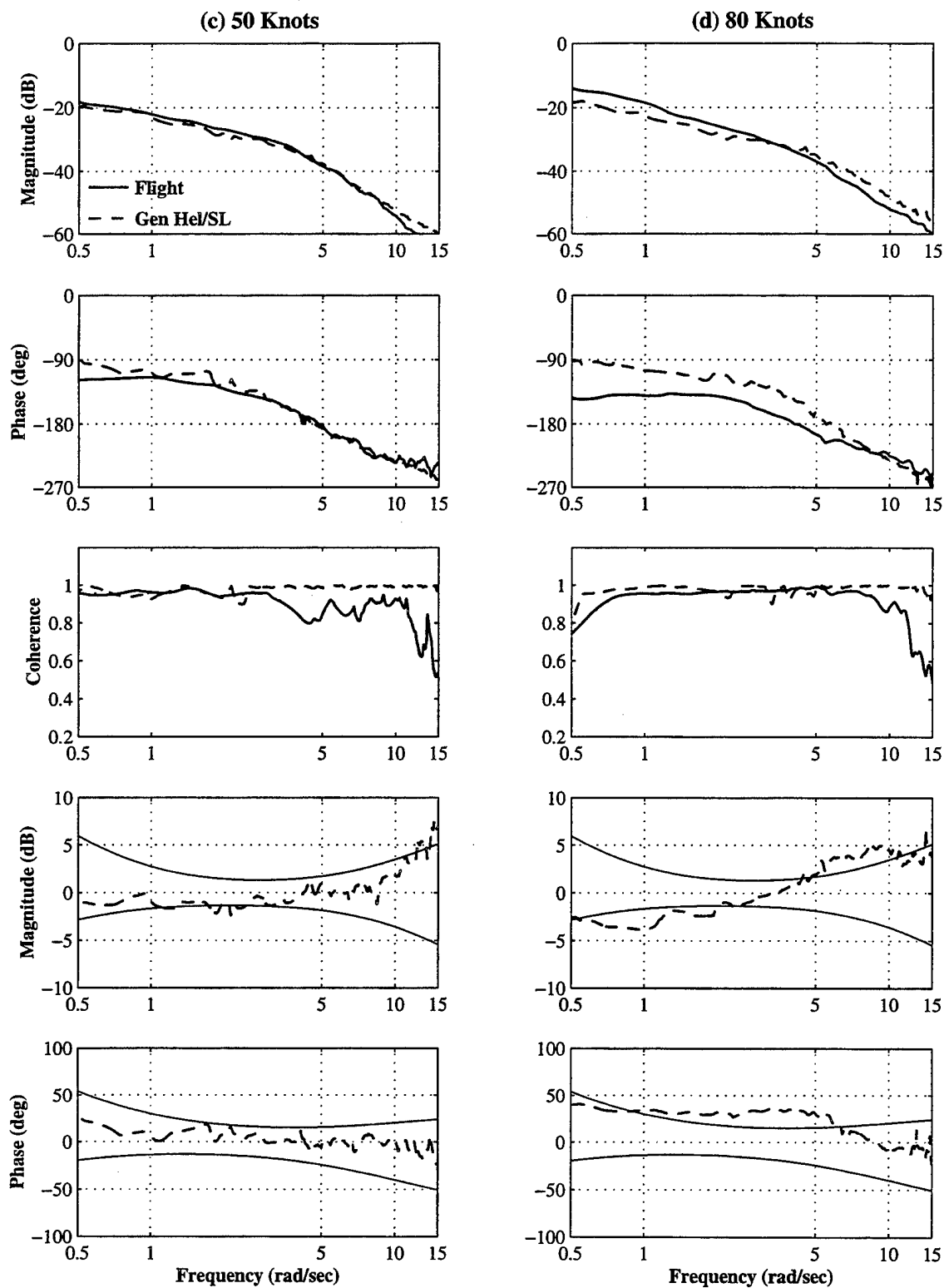


Figure D.3. Continued, (c) 50 Knots, (d) 80 Knots

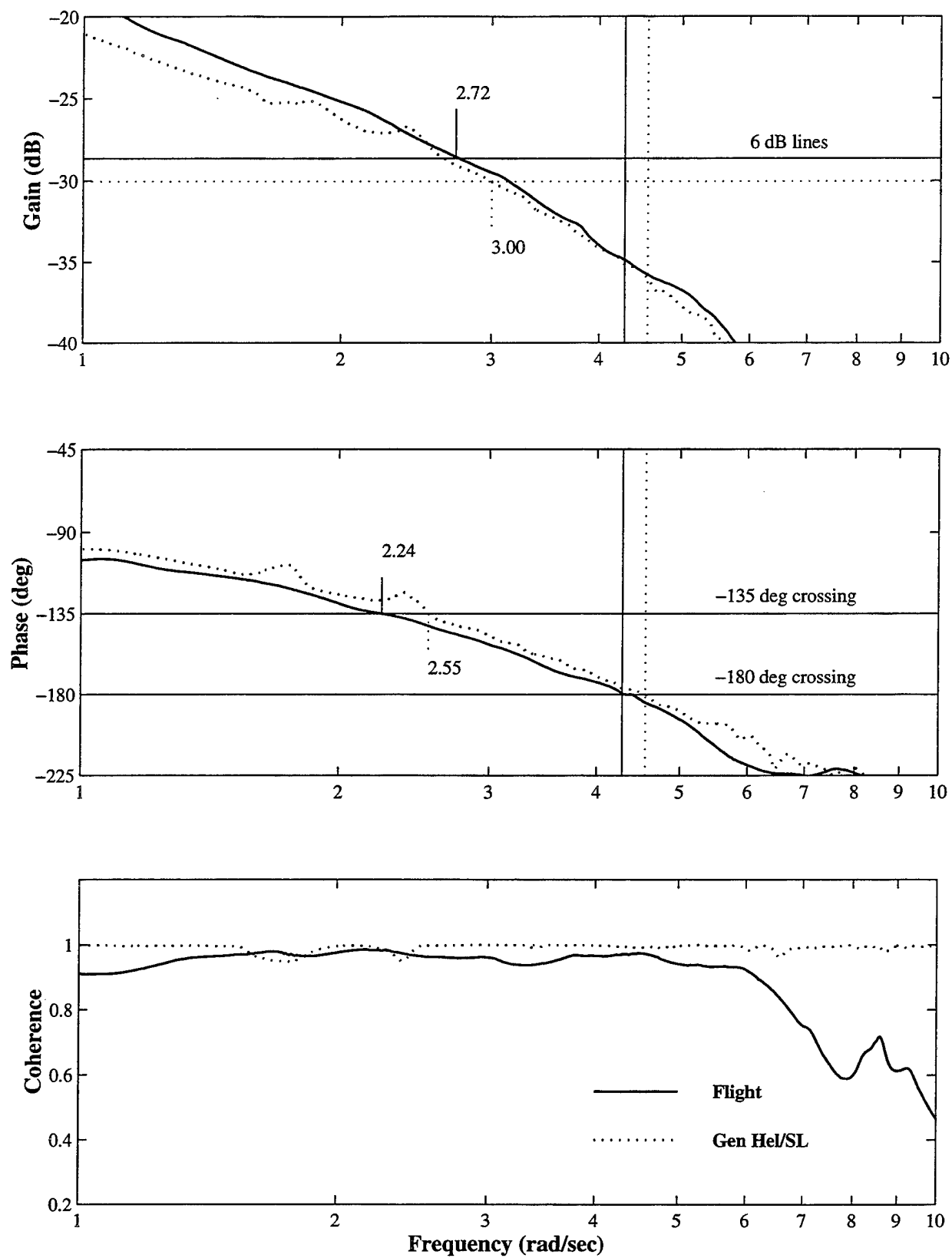


Figure D.4. No Load Handling Quality Determination, Longitudinal Axis, (a) Hover

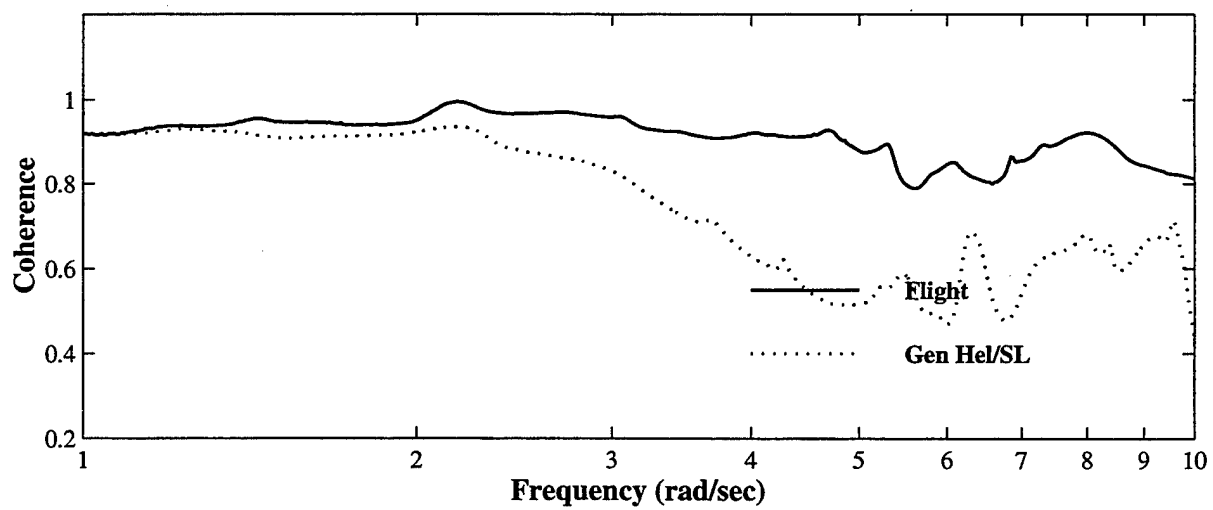
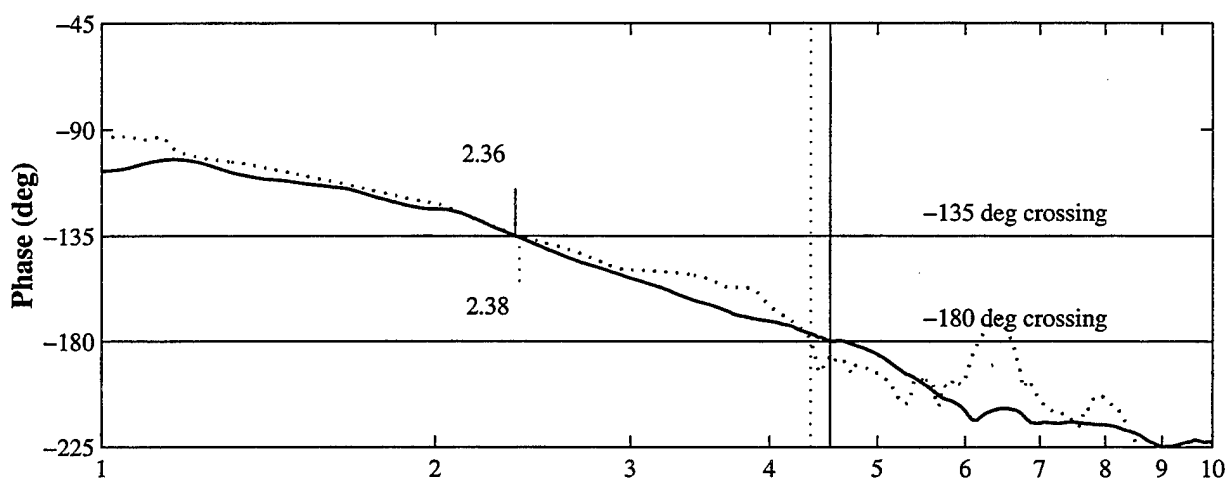
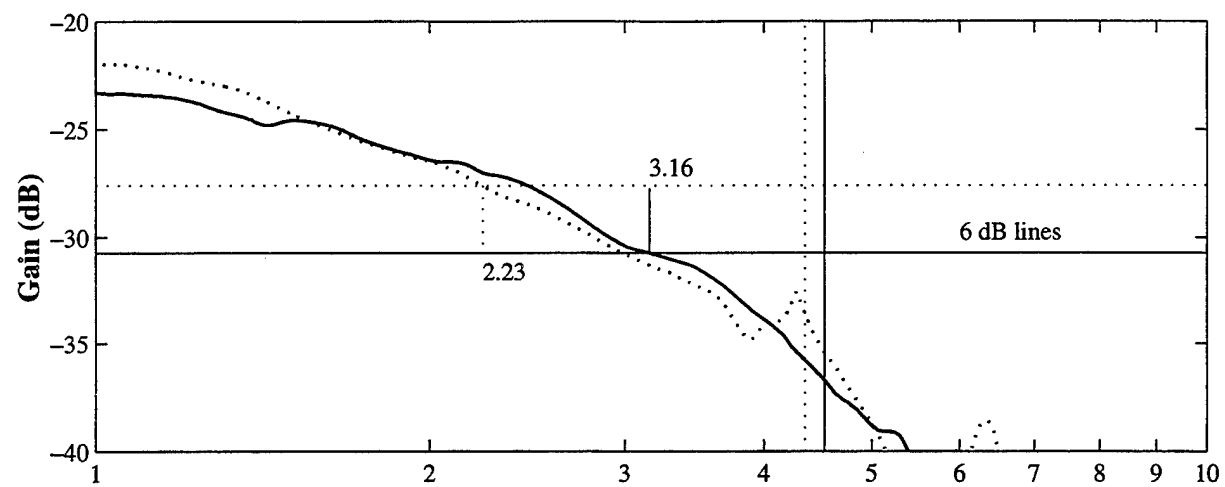


Figure D.4. Continued, (b) 30 Knots



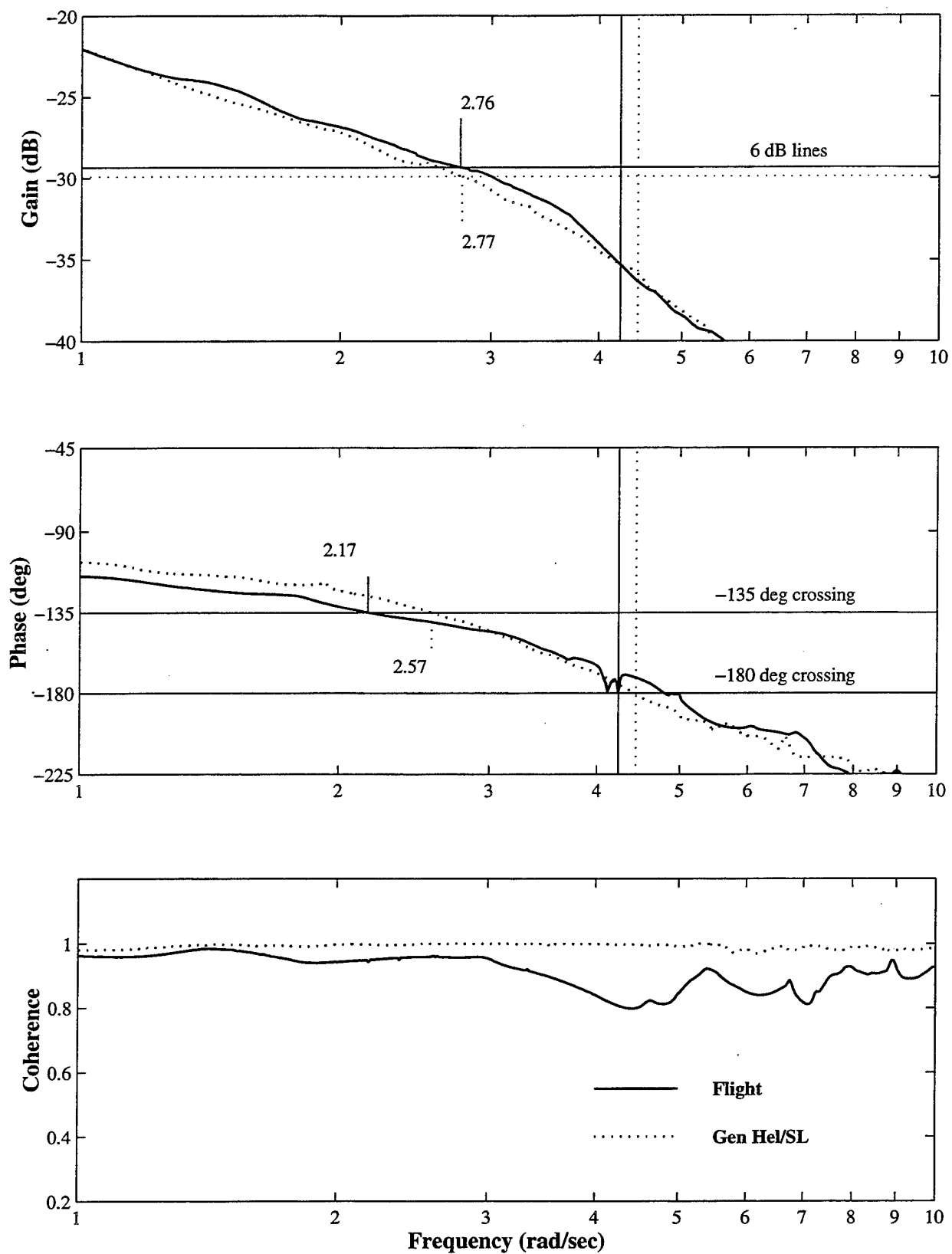


Figure D.4. Continued, (c) 50 Knots

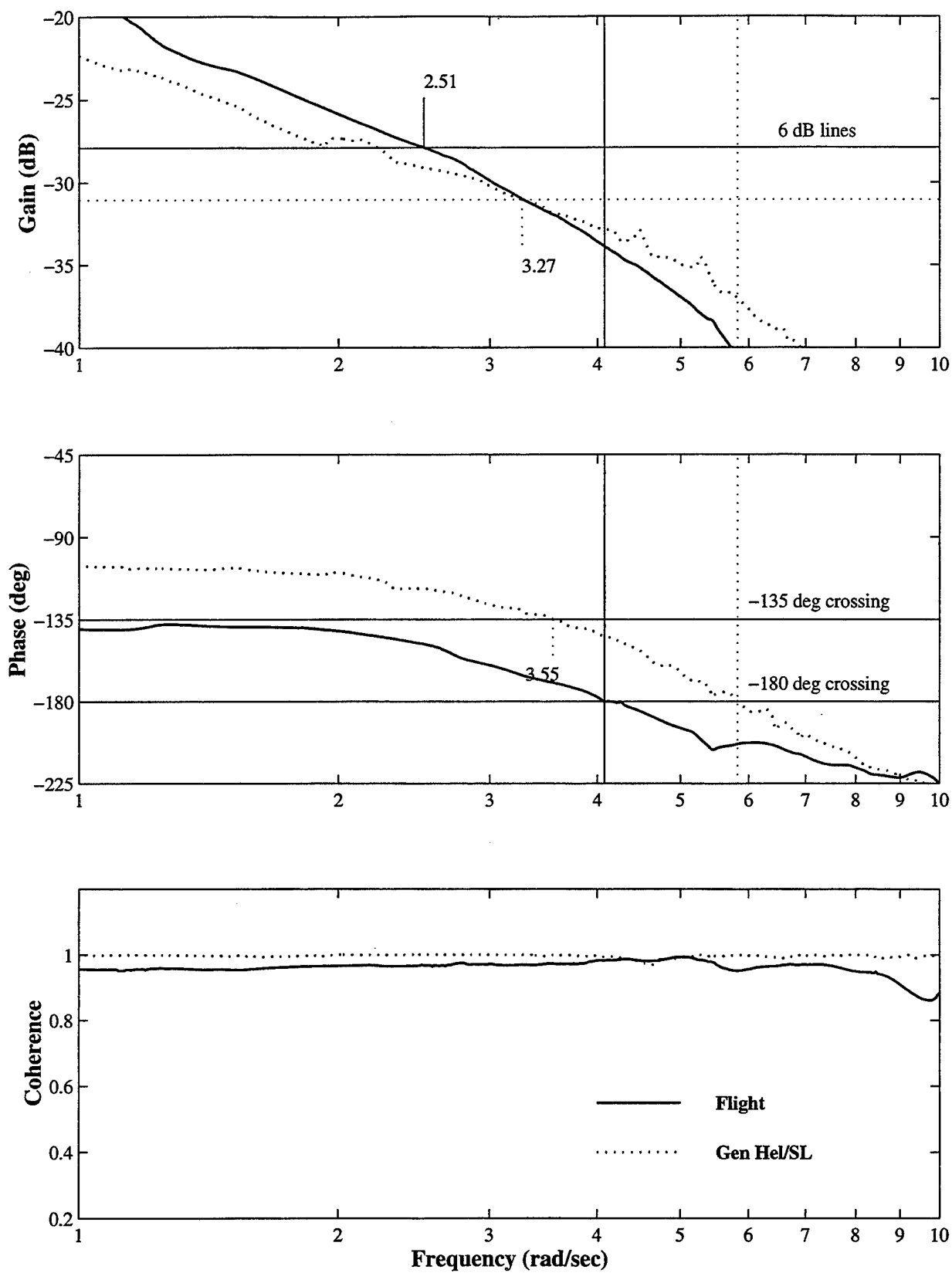


Figure D.4. Continued, (d) 80 Knots

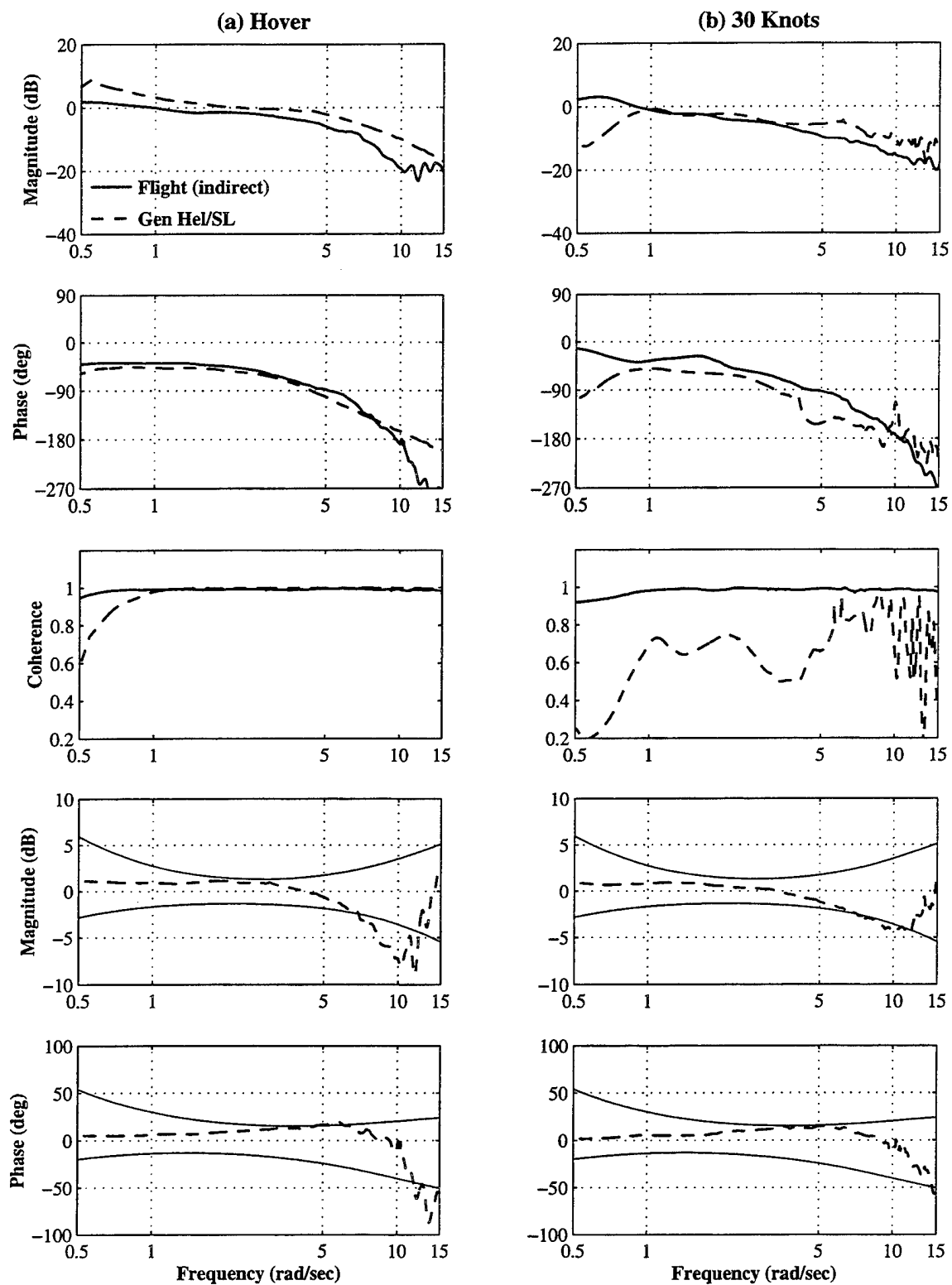


Figure D.5. No Load Stability Margin, Lateral Axis, (a) Hover, (b) 30 Knots

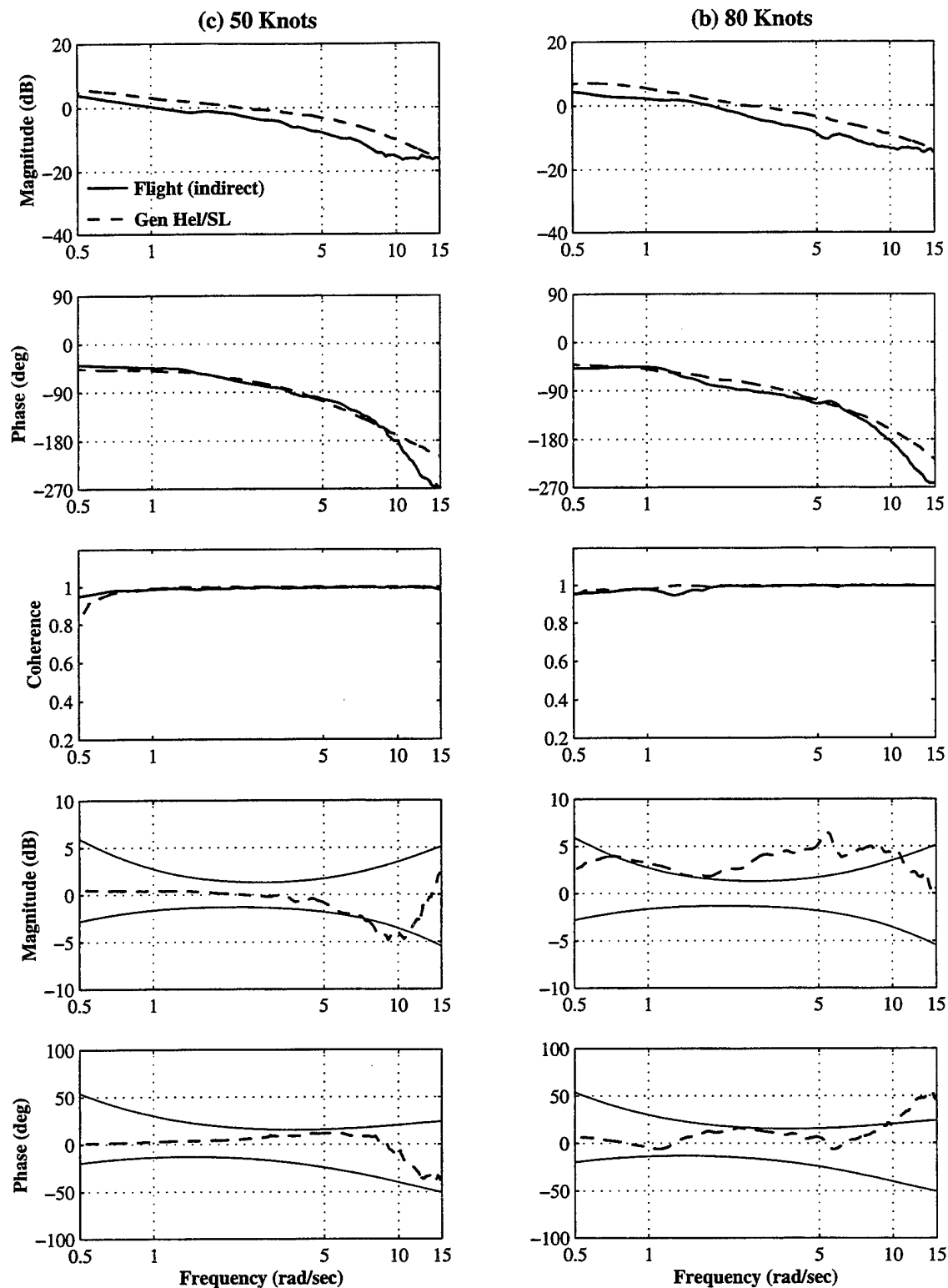


Figure D.5. Continued, (c) 50 Knots, (d) 80 Knots

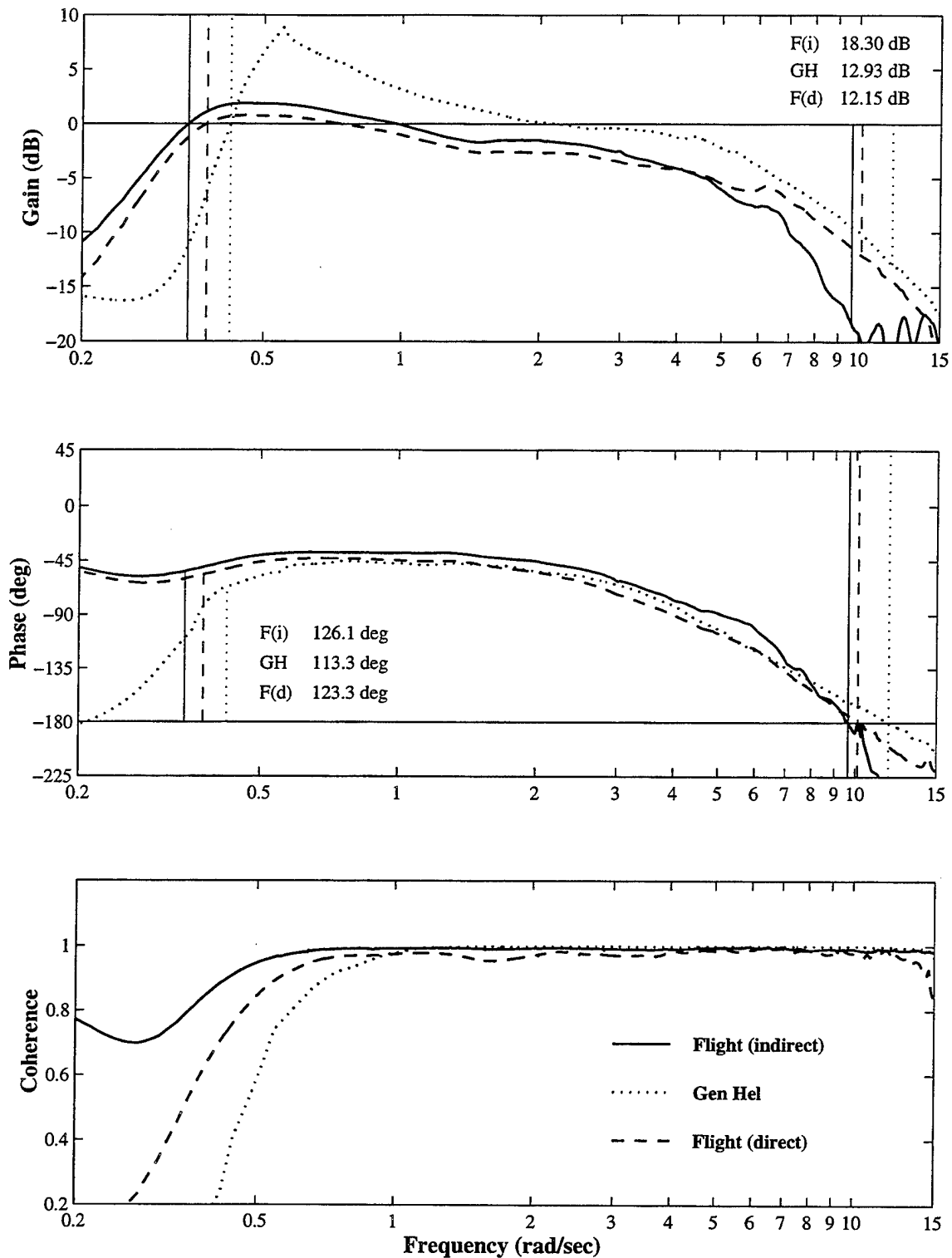


Figure D.6. No Load Stability Margin Determination, Lateral Axis, (a) Hover

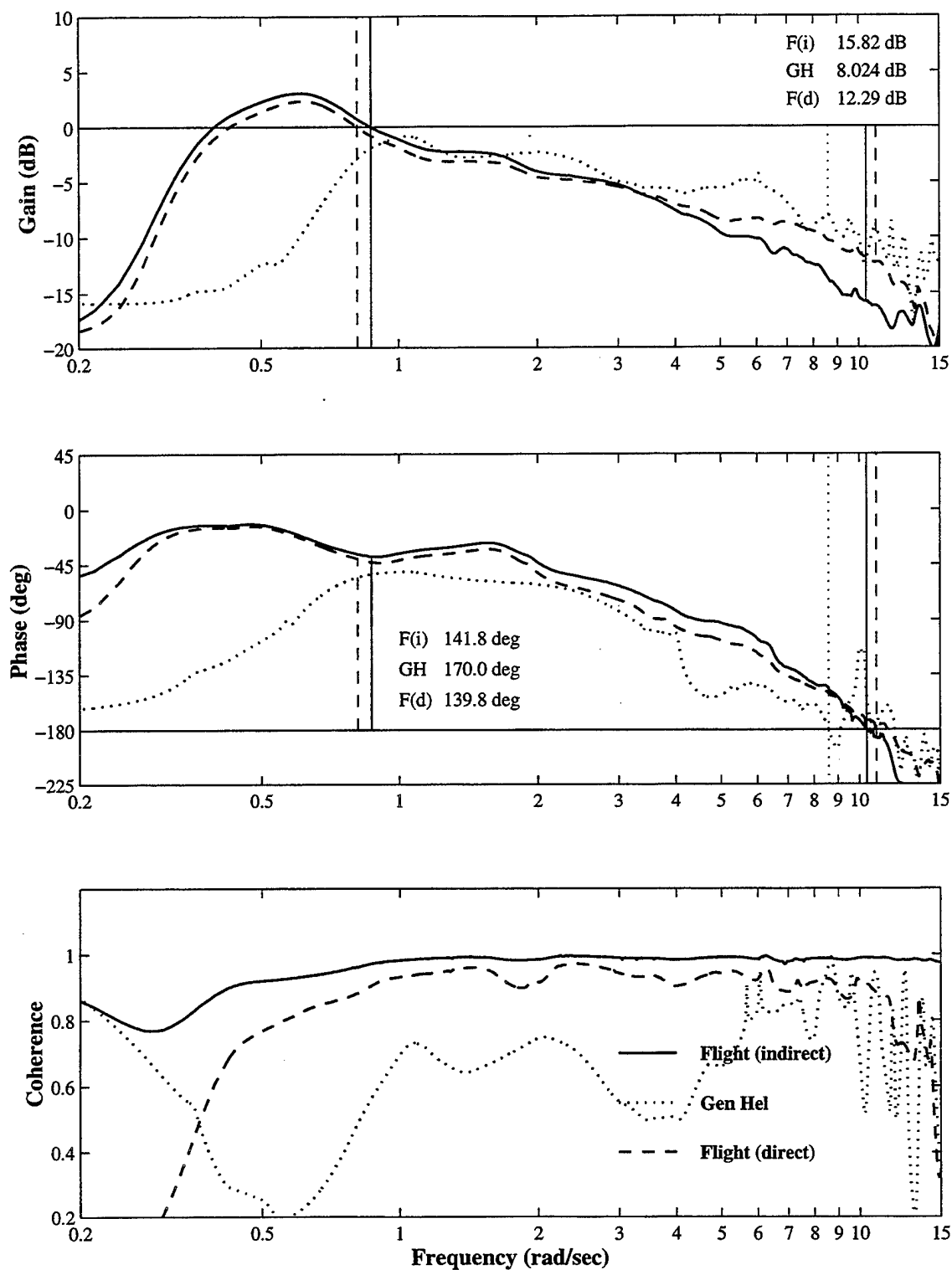


Figure D.6. Continued, (b) 30 Knots

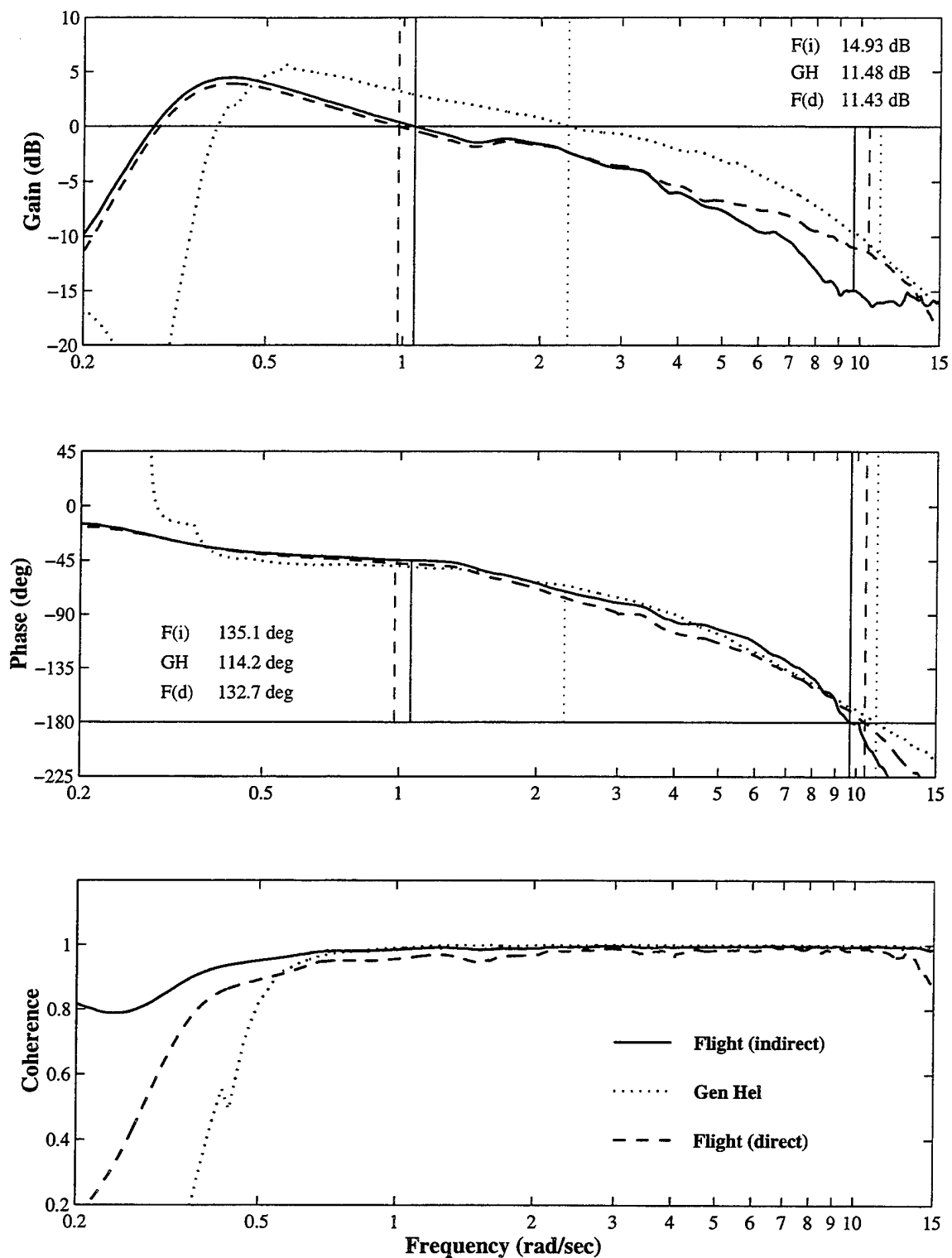


Figure D.6. Continued, (c) 50 Knots

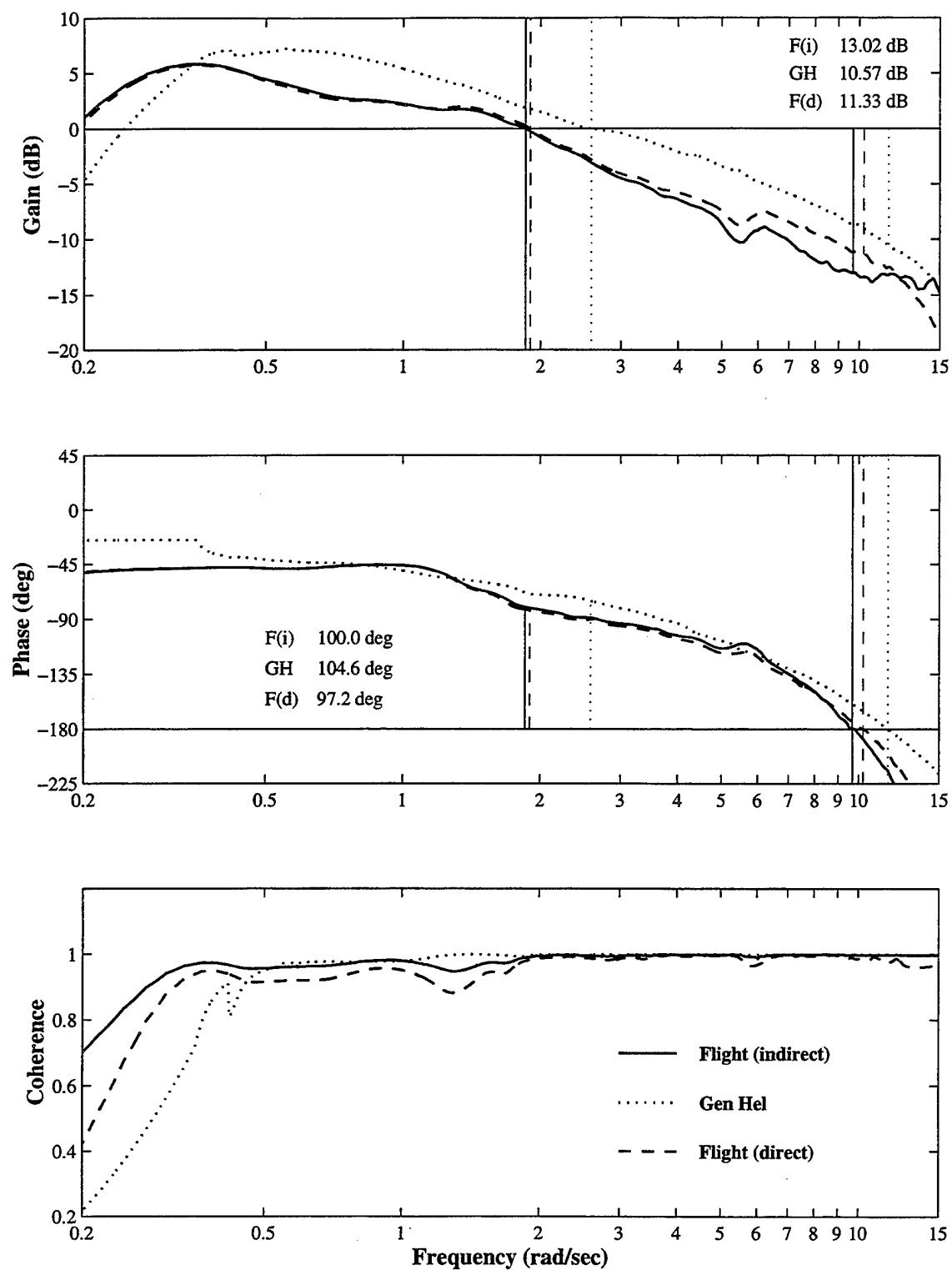


Figure D.6. Continued, (d) 80 Knots



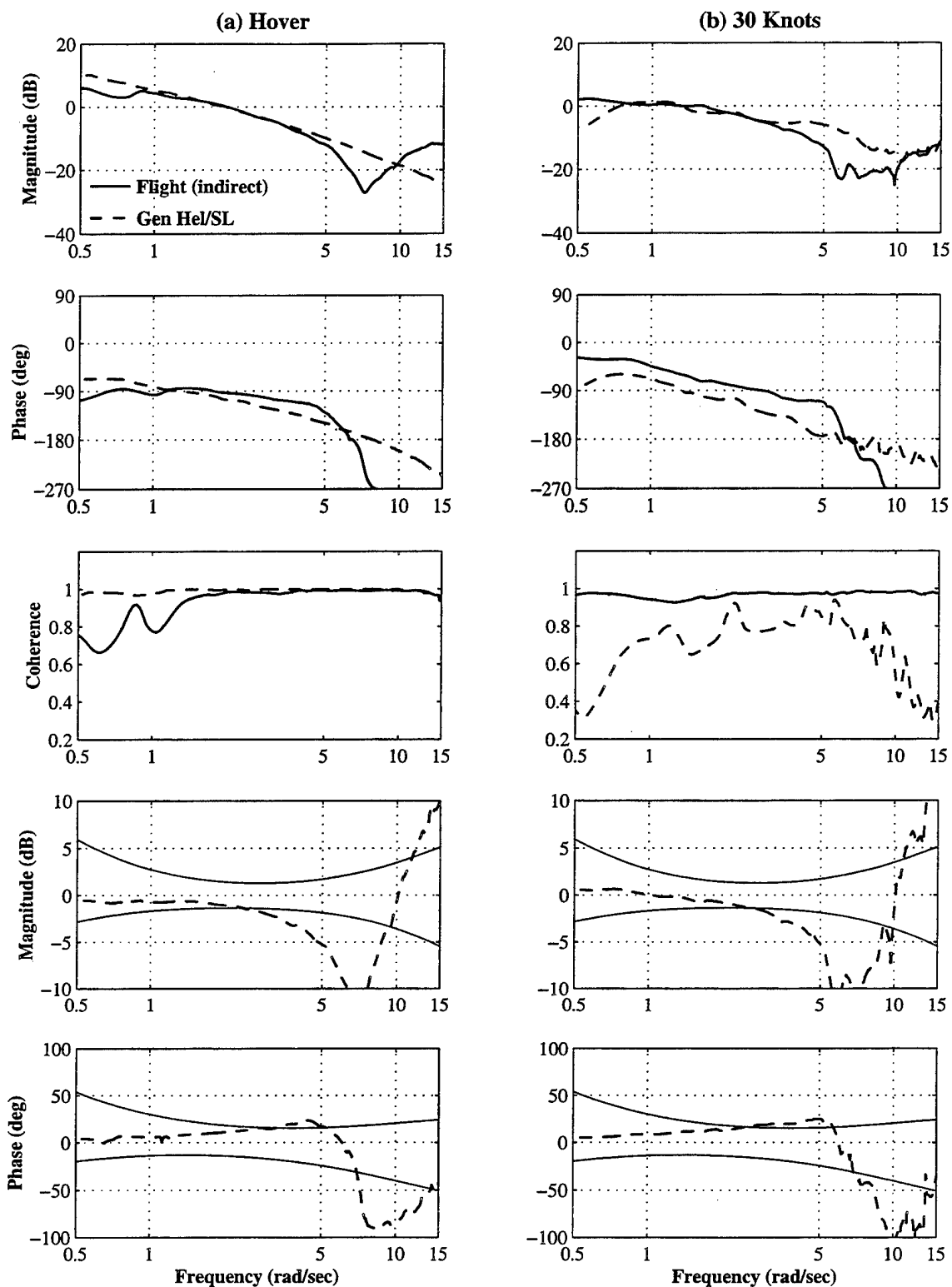


Figure D.7. No Load Stability Margin, Longitudinal Axis, (a) Hover, (b) 30 Knots

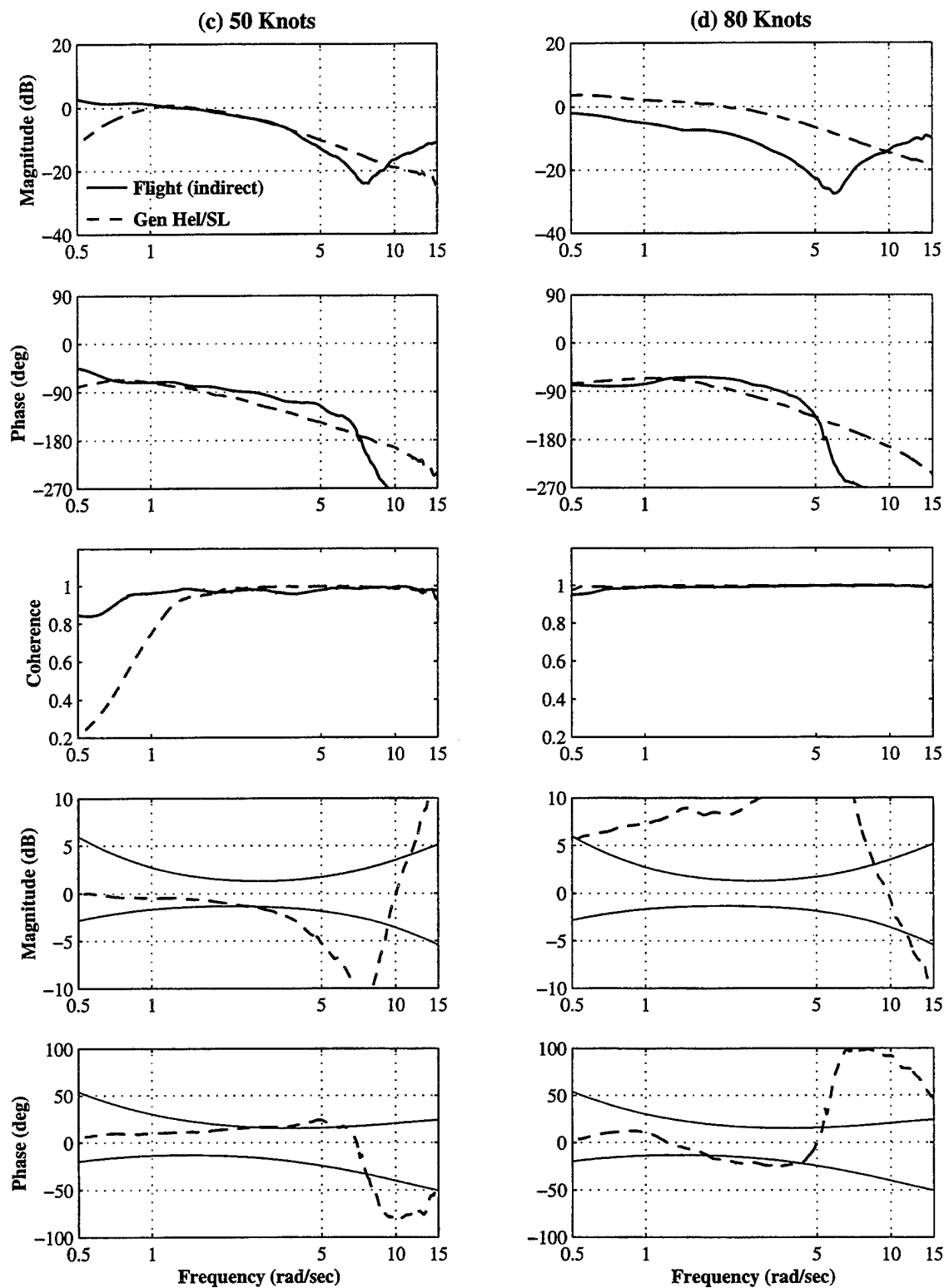


Figure D.7. Continued, (c) 50 Knots, (d) 80 Knots

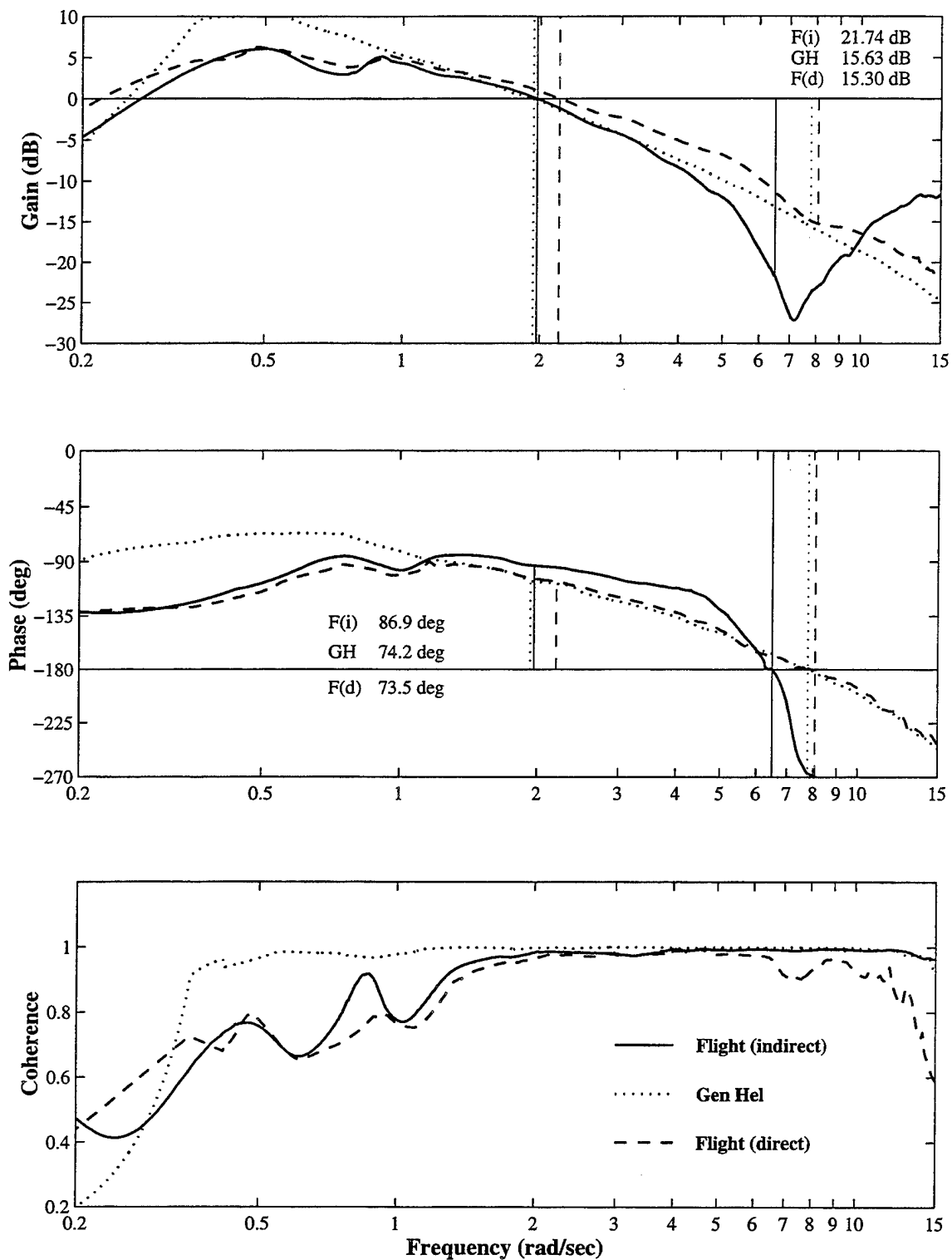


Figure D.8. No Load Stability Margin Determination, Longitudinal Axis, (a) Hover

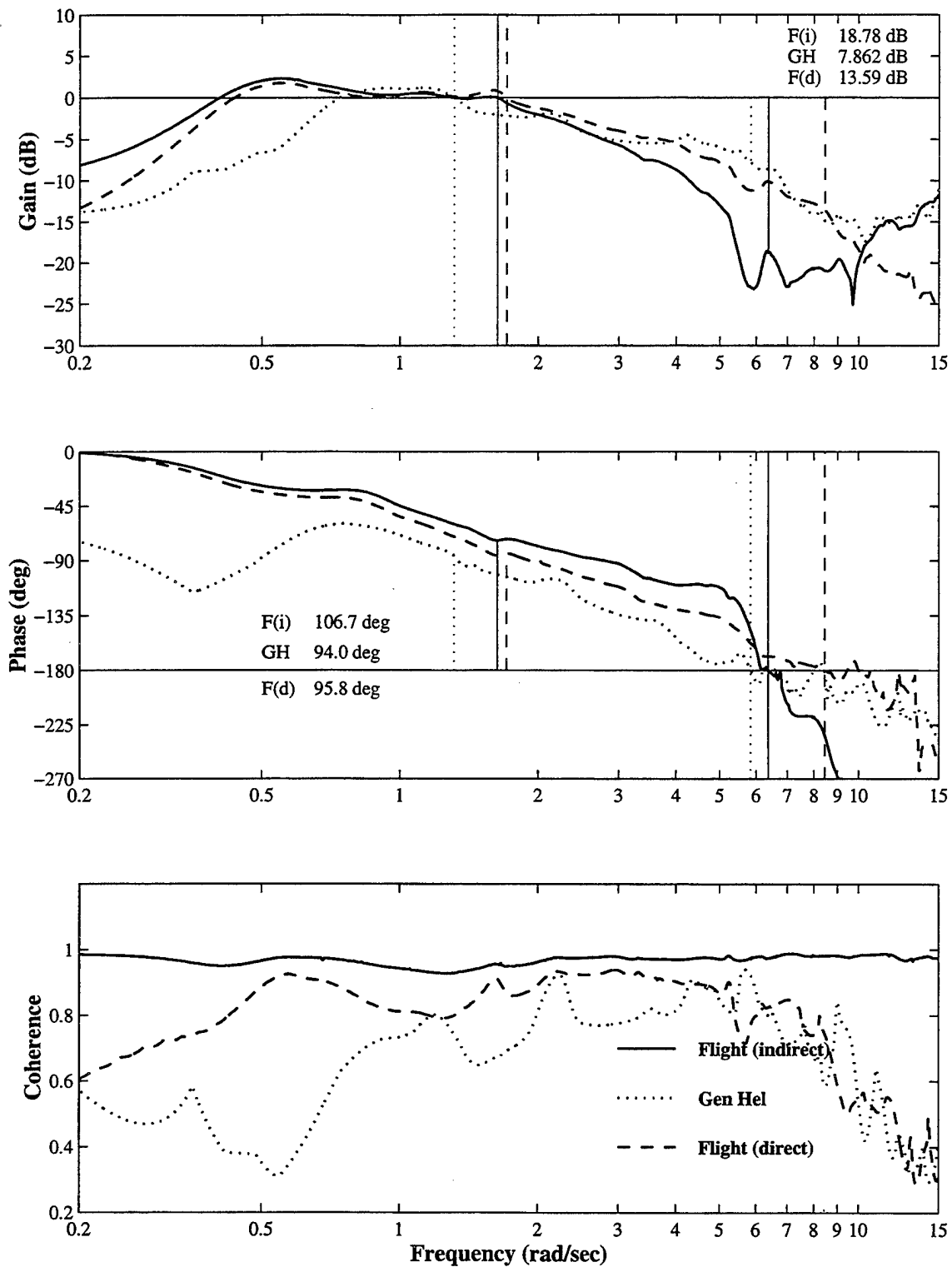


Figure D.8. Continued, (b) 30 Knots

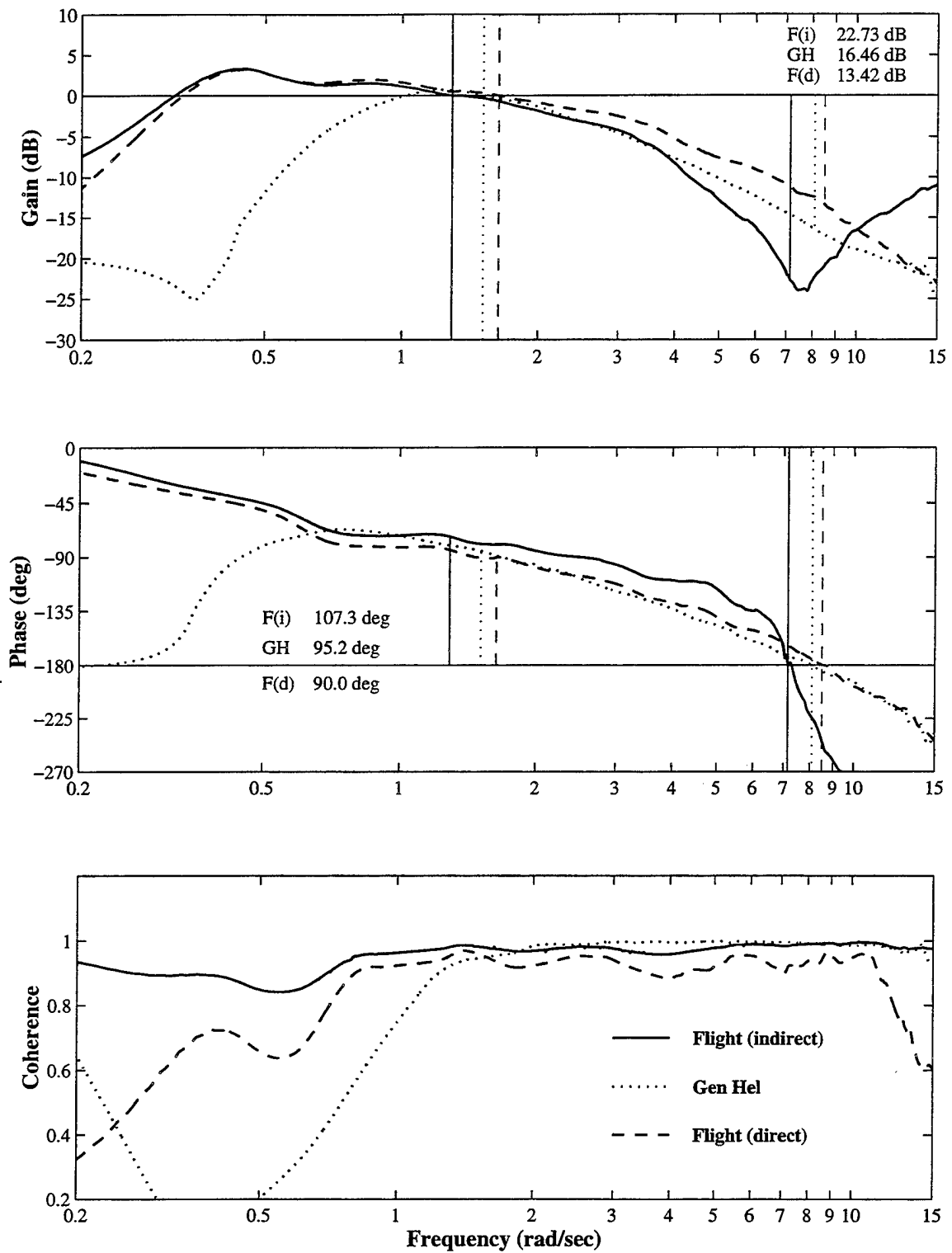


Figure D.8. Continued, (c) 50 Knots

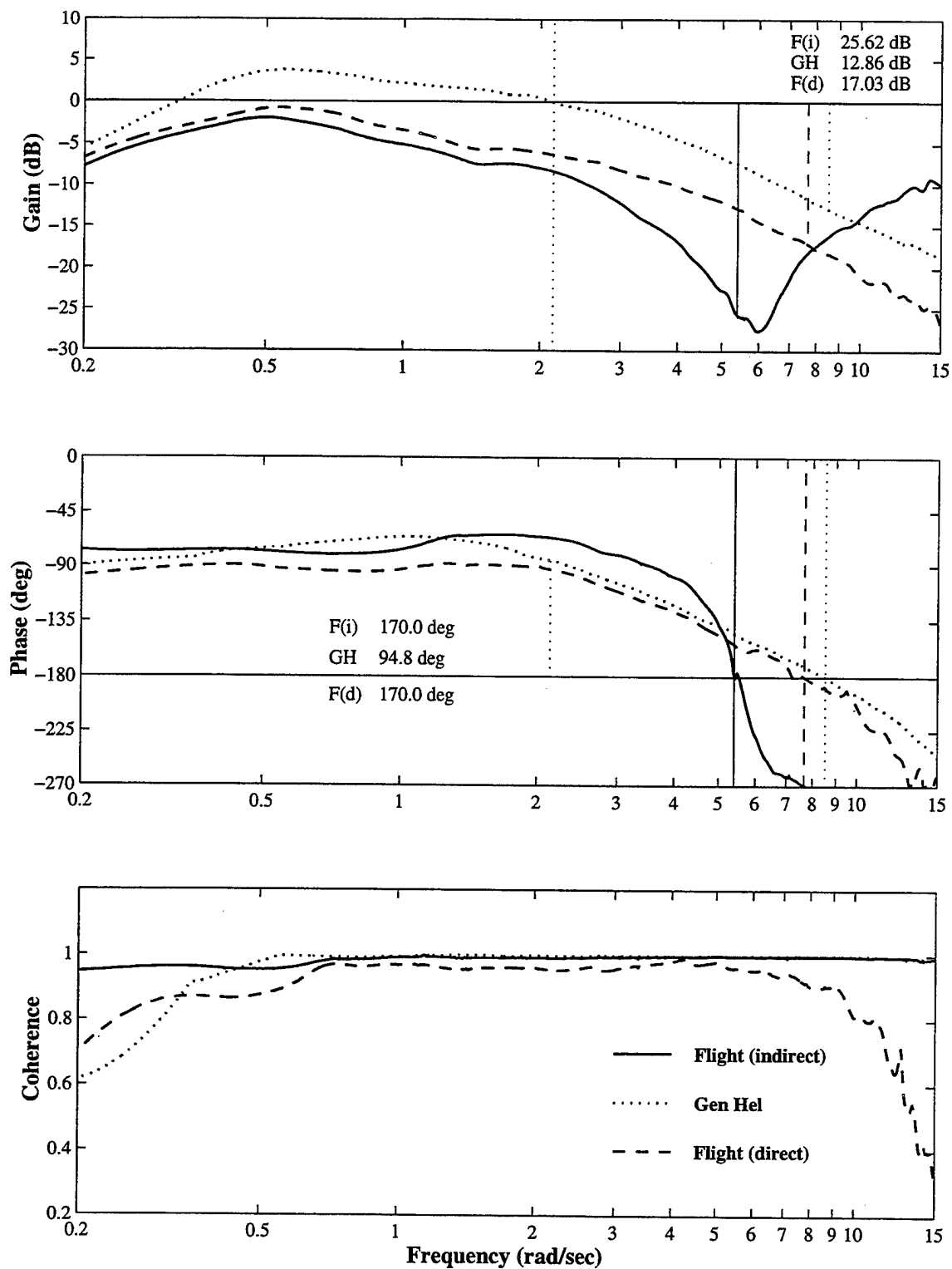


Figure D.8. Continued, (d) 80 Knots



## APPENDIX E. 4K BLOCK EXTERNAL LOAD DATA

This Appendix contains a summary of the data found for the 4K Block configuration from flight data as compared to Gen Hel/SL. In Table E.1, the cases presented at each airspeed for the lateral and longitudinal axes are given as (all Gen Hel/SL responses for HQ and SM are corrected with the average gain and time delay factors):

<b>Fi</b>	Flight, indirect method for computing Stability Margins
<b>Fd</b>	Flight, direct method for computing Stability Margins
<b>N</b>	Gen Hel/SL, no load aerodynamic forces and moments
<b>D</b>	Gen Hel/SL, drag only load aerodynamics estimation

Table E.2 shows the load motion characteristics as computed by NAVFIT over a frequency range of [0.5, 2.5] rad/sec.

Figures E.1 through E.4 show the helicopter attitude response to aircraft attitude as required by HQ analysis. Figures E.1 and E.3 present the overall frequency response across the range [0.5, 20] rad/sec and include magnitude and phase error functions with the proposed Level D criteria. Figures E.2 and E.4 give the same responses over the range [1, 10] rad/sec, providing a detailed view of the HQ parameter determination along with data coherence.

Figures E.5 through E.8 show the broken loop response used in the determination of SM from Flight, measured by both direct and indirect methods, and from the corrected Gen Hel/SL simulation results. Figures E.5 and E.7 provide the data for the range [0.5, 15] rad/sec, and include the error functions. Figures E.6 and E.8 provide details on SM determination.

Figures E.9 and E.11 present the load motion frequency responses from the Flight data and from the Gen Hel/SL simulation. The Gen Hel/SL data do not have any correction factor applied, and represent no load aerodynamics. The frequency responses, coherence, and error functions are shown over the range [0.5 5] rad/sec.



Case			-180° Crossing (rad/s)	-180° Gain (dB)	-135° Band- width (rad/s)	6 dB Band- width (rad/s)	Phase Delay (s)	LS Fit Range (rad/s)	Gain Margin (dB)	Gain Crossing (rad/s)	Phase Margin (deg)	Phase Crossing (rad/s)
L A T E R A L	0	Fi	8.481	-32.11	6.505	3.835	0.1299	6-10	12.11	10.75	108.7 101.8	0.7154 0.9712
		Fd							10.94	10.59	-	-
		N	7.655	-32.82	5.384	4.044	0.1391		13.44	12.00	89.77 150.0 117.1	1.056 2.497 3.599
		D	7.655	-32.85	5.385	4.069	0.1386		13.45	12.02	90.72 150.5 120.1	1.057 2.505 3.530
	3 0	Fi	7.922	-31.19	5.349	4.207	0.1533	6-10	16.46	9.941	109.9	0.9863
		Fd							10.40	10.55	90.07	0.9546
		N	7.679	-32.42	5.413	3.634	0.1477		13.12	11.69	90.81 149.1 129.1	1.139 2.484 3.129
		D	7.705	-32.33	5.407	3.602	0.1485		12.35	11.38	92.67 149.7 129.6	1.140 2.493 3.114
	5 0	Fi	8.151	-31.95	5.536	4.311	0.1517	6-10	15.17	9.845	109.0 157.9 134.5	1.009 2.079 2.760
		Fd							10.86	11.11	86.11 90.04 146.5 131.8	0.6655 0.8563 2.374 2.663
		N	7.655	-32.39	5.330	3.778	0.1437		11.38	11.03	84.91 153.7 116.7	1.139 2.298 3.260
		D	7.655	-32.40	5.330	3.790	0.1436		11.39	11.04	85.27 153.3 116.7	1.138 2.308 3.262
	8 0	Fi	8.520	-31.85	6.317	4.112	0.1297	6-10	10.86	10.15	96.96	1.084
		Fd							9.410	10.43	93.66	1.084
		N	7.985	-33.32	5.278	4.211	0.1214		10.88	11.49	85.84 179.3 111.1	1.158 1.996 3.260
		D	7.992	-33.33	4.815	4.220	0.1219		10.91	11.48	85.85 178.8 111.0	1.158 2.006 3.254

Table E.1. Helicopter Response Summary: 4K Block Load Configuration, Lateral Axis

Case			-180° Crossing (rad/s)	-180° Gain (dB)	-135° Band- width (rad/s)	6 dB Band- width (rad/s)	Phase Delay (s)	LS Fit Range (rad/s)	Gain Margin (dB)	Gain Crossing (rad/s)	Phase Margin (deg)	Phase Crossing (rad/s)
LONGITUDINAL	0	Fi	4.686	-34.85	2.689	3.057	0.1940	4-6	16.93	7.184	90.22 91.10 82.62	1.399 1.818 2.411
		Fd							-	-	-	-
		N	4.623	-35.99	2.608	3.129	0.2050		15.53	7.910	82.51 79.71 76.65	1.279 1.869 2.198
		D	4.623	-35.93	2.608	3.114	0.2055		15.52	7.905	82.61 79.61 76.69	1.287 1.873 2.203
	30	Fi	5.022	-35.96	2.836	3.292	0.1694	4-6	16.92	7.182	128.2 96.43	0.711 1.301
		Fd							-	-	-	-
		N	4.584	-35.89	2.854	3.089	0.2282		15.43	7.913	-	-
		D	4.806	-35.87	2.962	2.974	0.2166		15.39	7.910	89.63 95.35 84.70	1.290 1.865 2.247
	50	Fi	5.217	-35.96	2.829	3.159	0.1654	4-6	15.97	7.309	98.96 108.2 98.47	1.246 1.862 2.148
		Fd							-	-	-	-
		N	4.860	-36.85	3.104	3.265	0.1986		14.88	8.212	93.67 102.5 92.26	1.262 1.864 2.183
		D	4.863	-36.50	3.155	3.160	0.1986		14.72	8.227	94.75 106.7 93.11	1.263 1.857 2.189
	80	Fi	5.170	-34.44	3.281	2.521	0.1718	4-6	13.71	7.156	99.42	1.065
		Fd							10.03	7.902	90.60 86.73	1.453 2.797
		N	5.733	-36.57	3.704	3.086	0.1820		12.98	8.342	105.9 124.3 102.4	1.338 1.887 2.302
		D	5.740	-36.52	3.682	3.063	0.1824		12.95	8.353	105.6 129.3 103.2	1.337 1.882 2.298

Table E.1. Continued, Longitudinal Axis



Case			Damping Ratio	Natural Frequency	Cost Function
L A T E R A L	0	F	0.1369	1.5632	99.09
		N	0.1371	1.6149	129.2
		D	0.1353	1.6255	117.4
	30	F	0.1960	1.4912	150.2
		N	0.1252	1.6110	129.8
		D	0.1239	1.6346	115.2
	50	F	0.1930	1.5531	101.6
		N	0.1328	1.6230	155.4
		D	0.1344	1.6277	151.6
	80	F	-	-	-
		N	0.1143	1.6347	175.9
		D	0.1160	1.6426	167.2
L O N G I T U D I N A L	0	F	0.0731	1.6568	49.74
		N	0.0663	1.6554	62.24
		D	0.0676	1.6651	59.38
	30	F	0.0808	1.5875	89.33
		N	0.0698	1.6891	40.58
		D	0.0668	1.6728	74.76
	50	F	0.1069	1.6041	59.54
		N	0.0622	1.6517	85.77
		D	0.0641	1.6559	85.99
	80	F	-	-	-
		N	0.0615	1.6586	83.92
		D	0.0672	1.6692	84.89

Table E.2. Load Motion Parameters: 4K Block Load Configuration

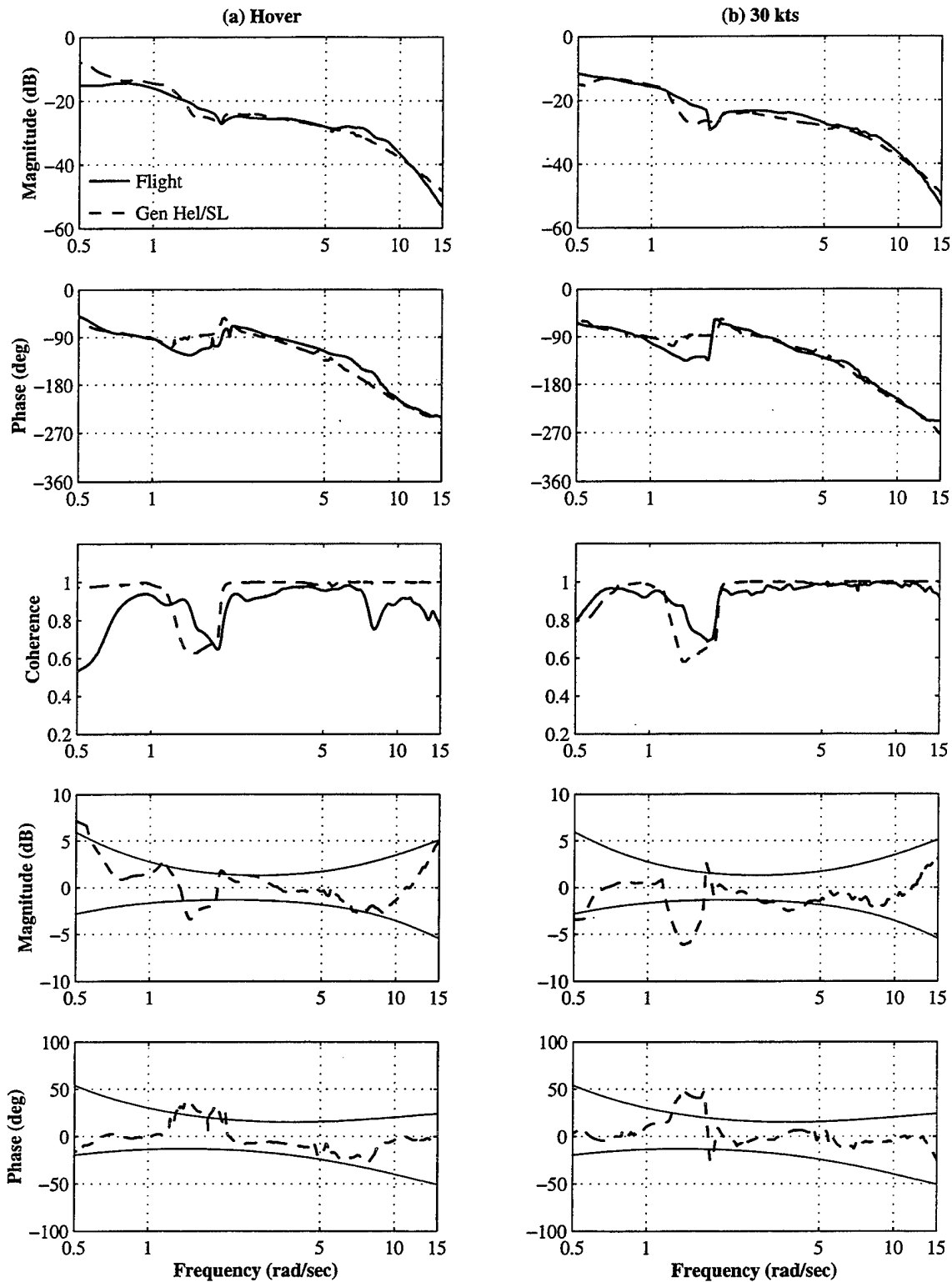


Figure E.1. 4K Block Handling Quality, Lateral Axis, (a) Hover, (b) 30 Knots

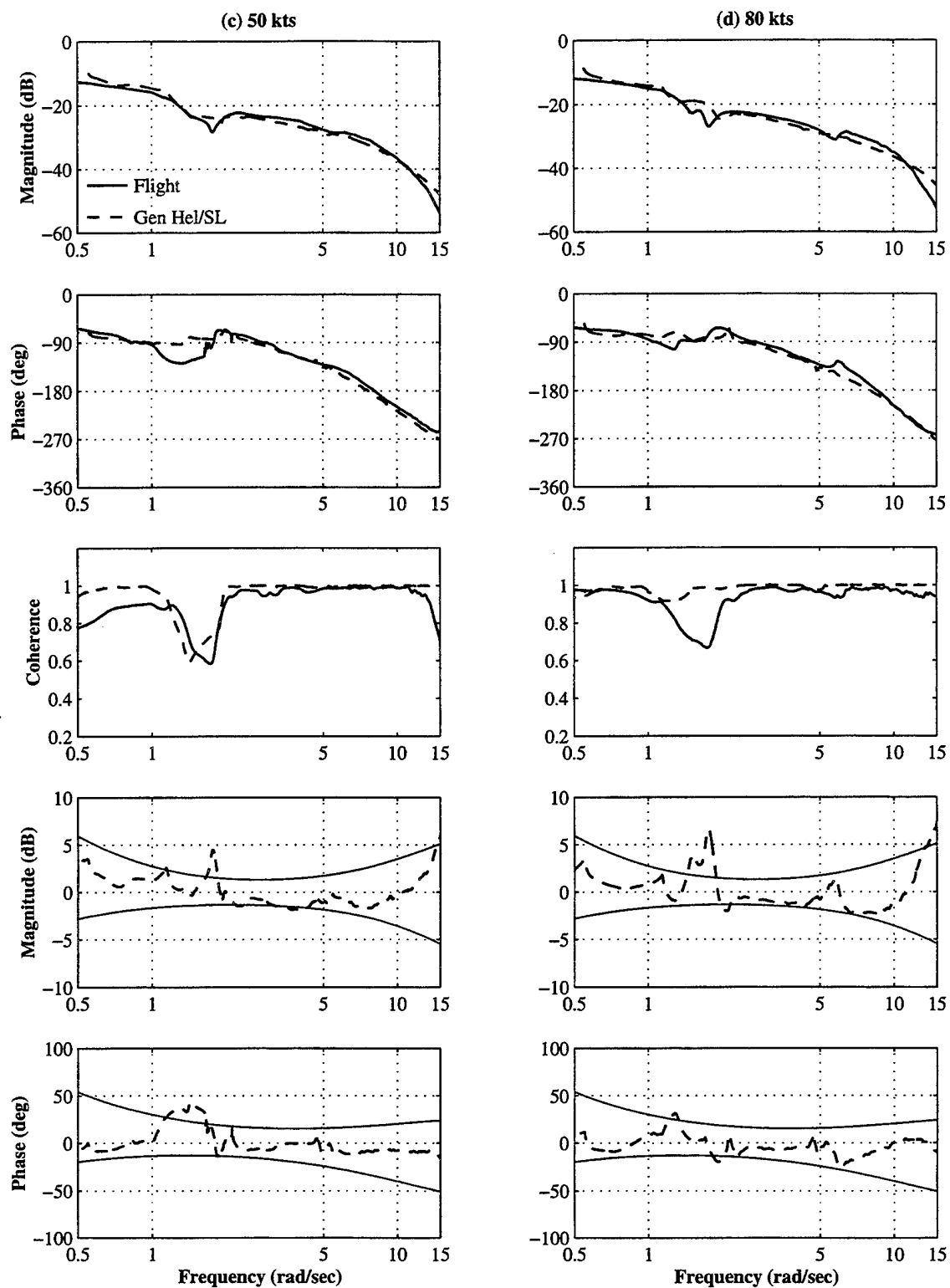


Figure E.1. Continued, (c) 50 Knots, (d) 80 Knots

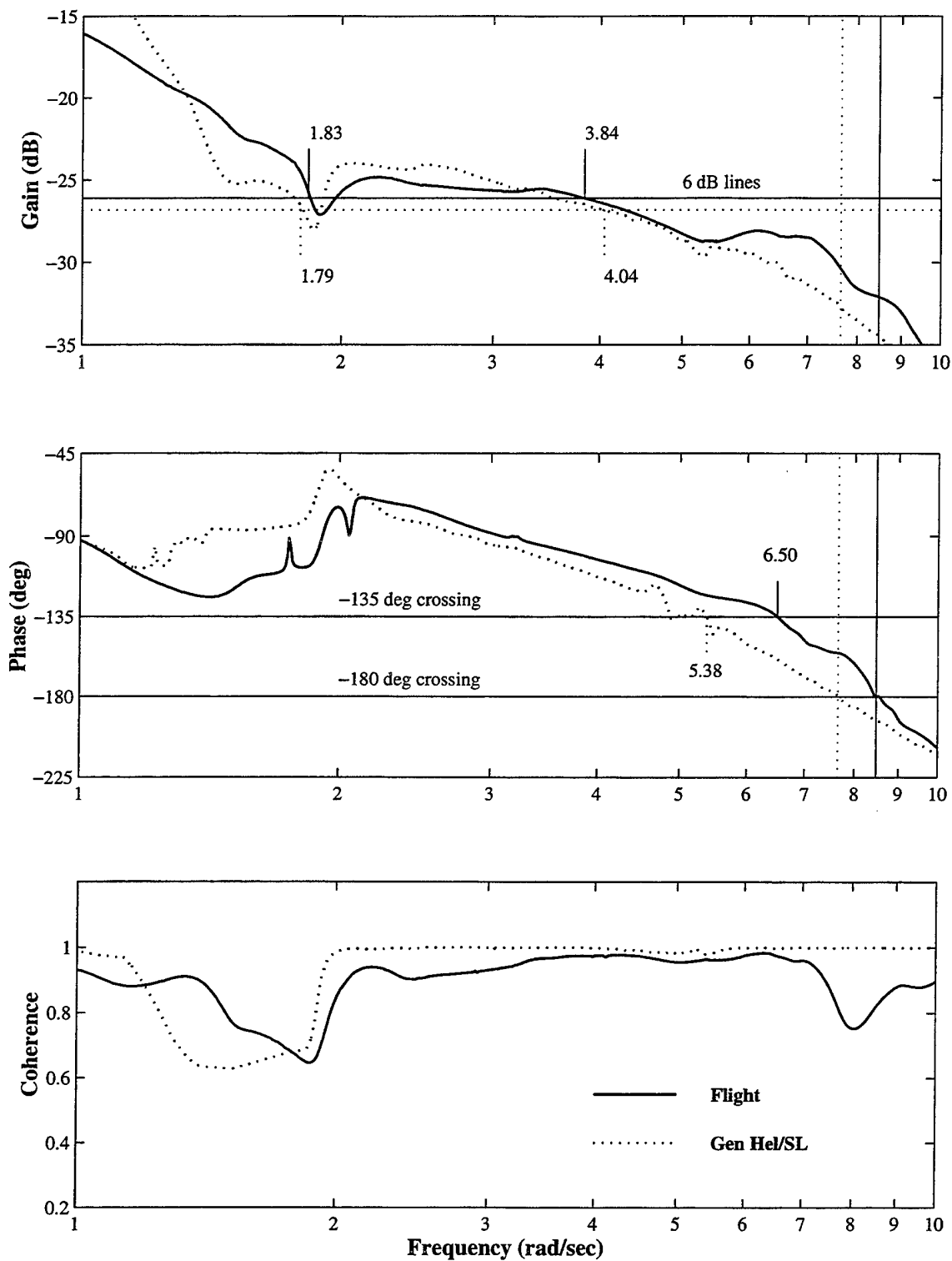


Figure E.2. 4K Block Handling Quality Determination, Lateral Axis, (a) Hover

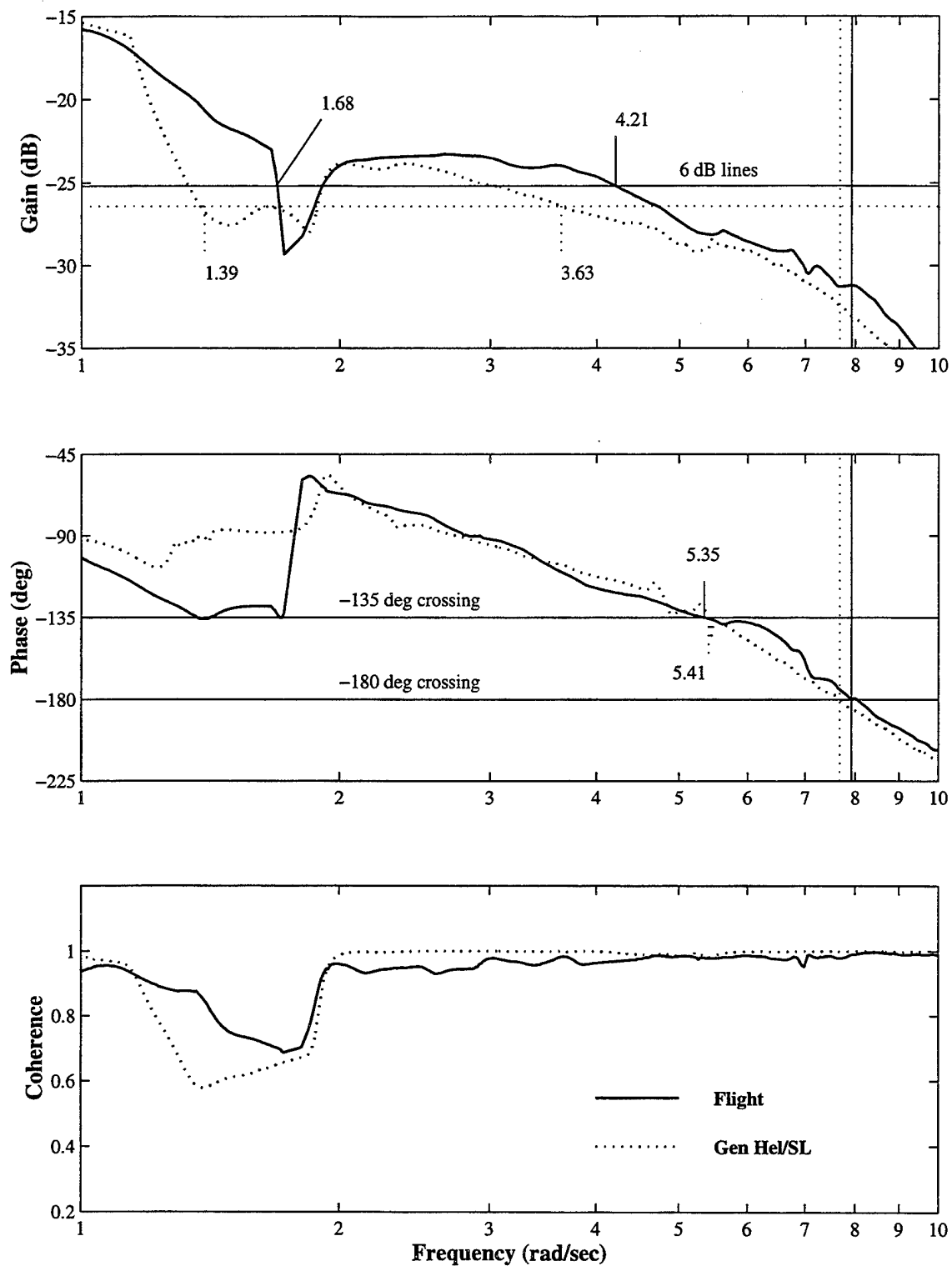


Figure E.2. Continued, (b) 30 Knots



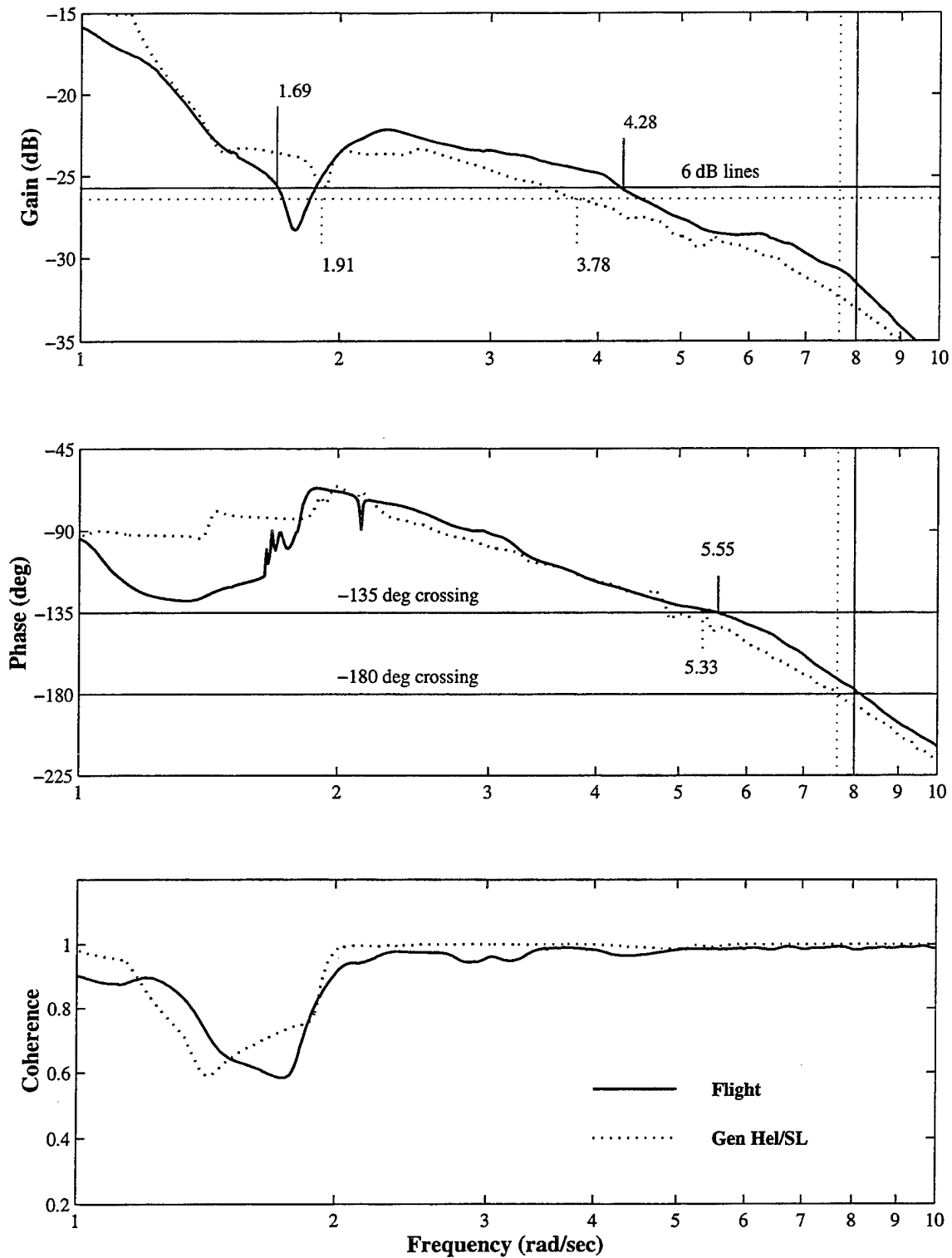


Figure E.2. Continued, (c) 50 Knots

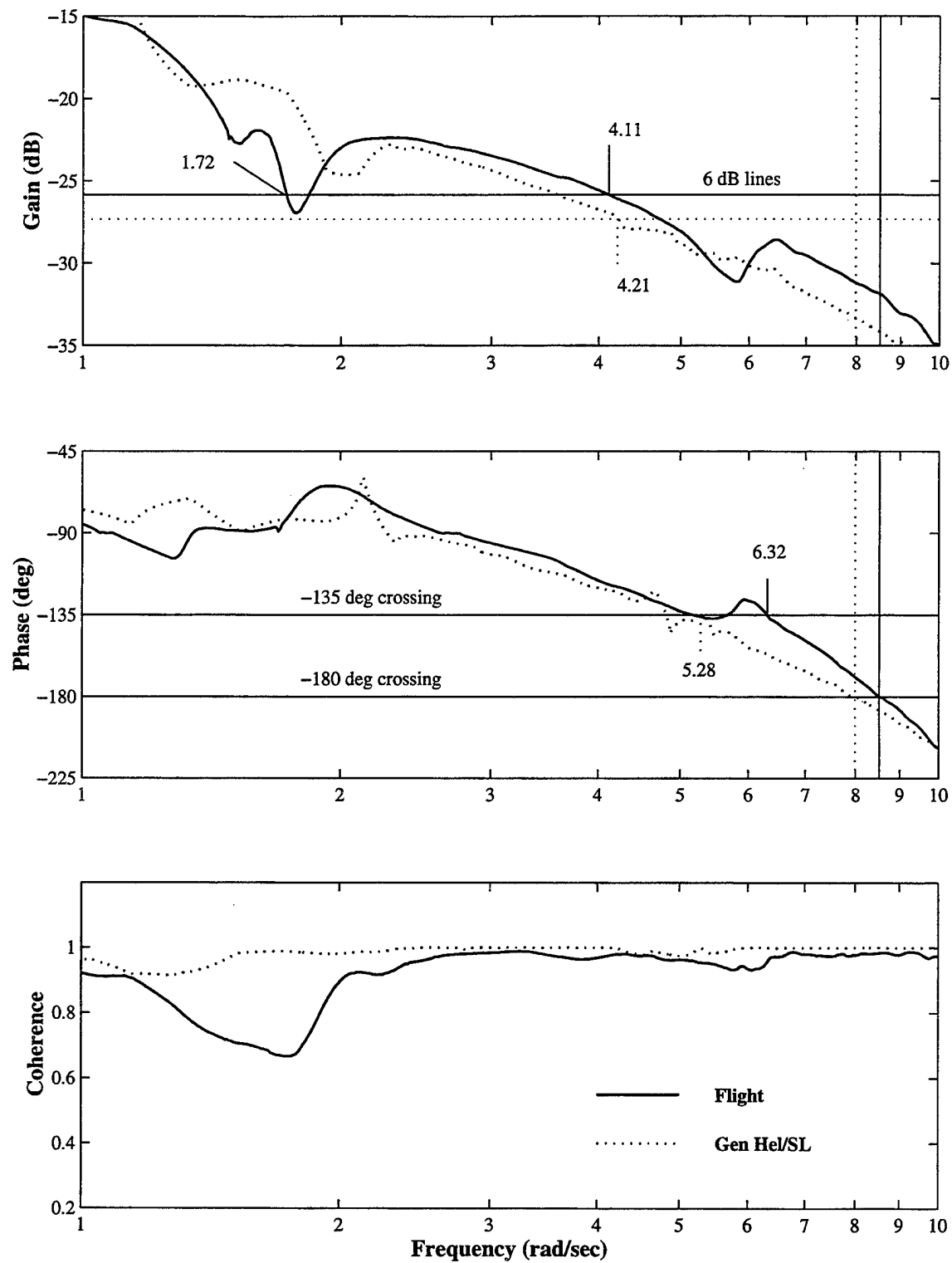


Figure E.2. Continued, (d) 80 Knots

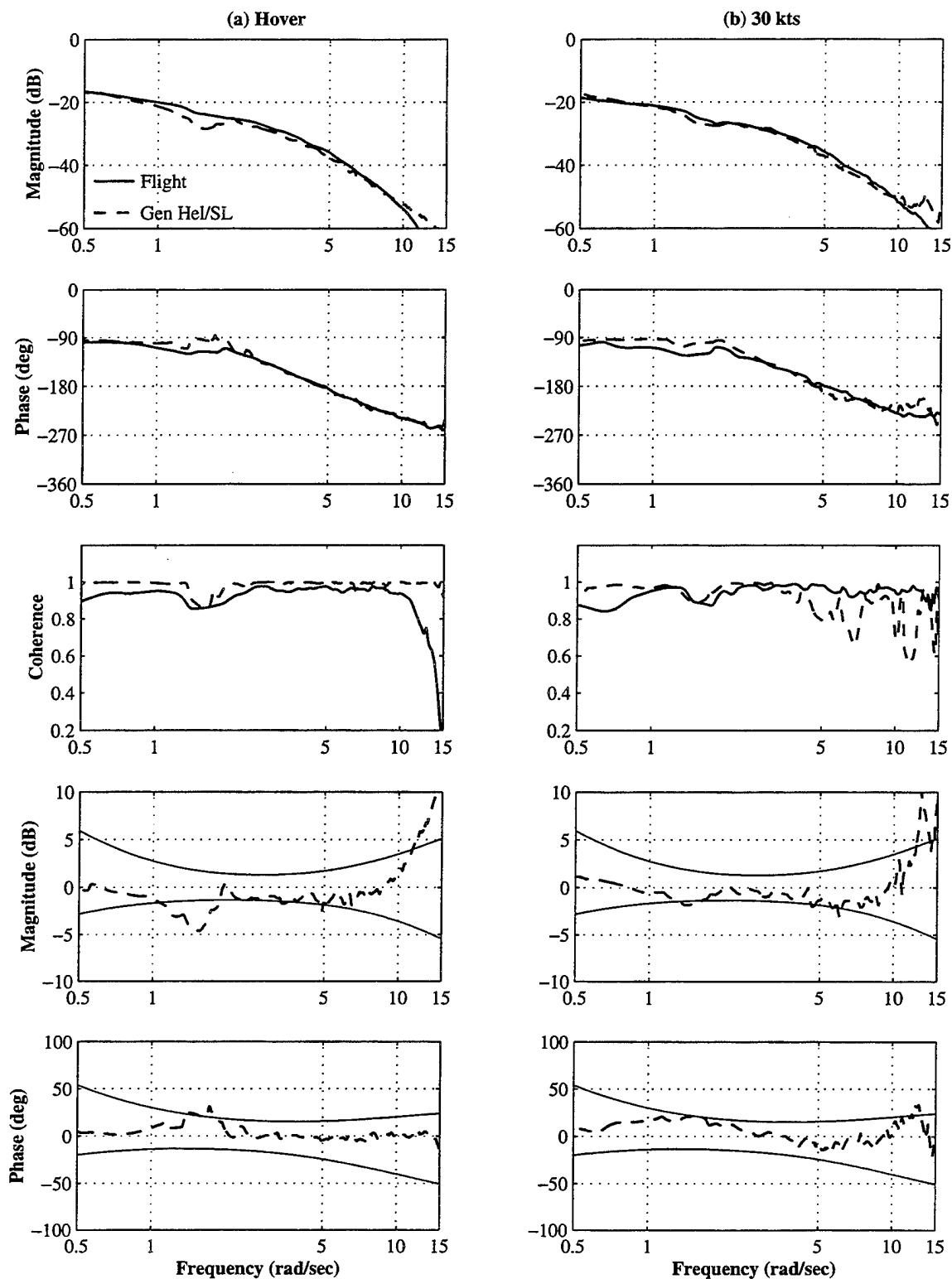


Figure E.3. 4K Block Handling Quality, Longitudinal Axis, (a) Hover, (b) 30 Knots

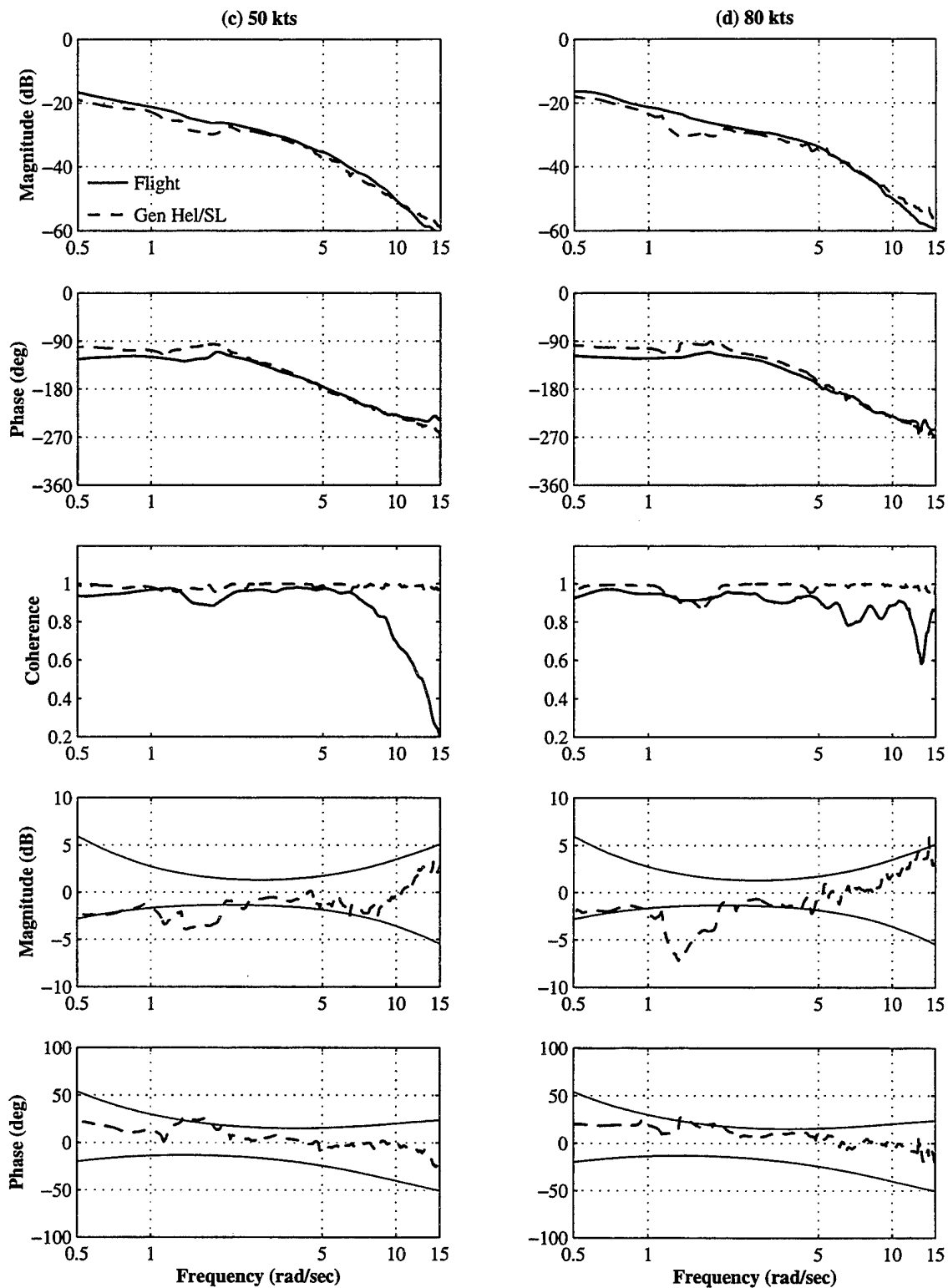


Figure E.3. Continued, (c) 50 Knots, (d) 80 Knots

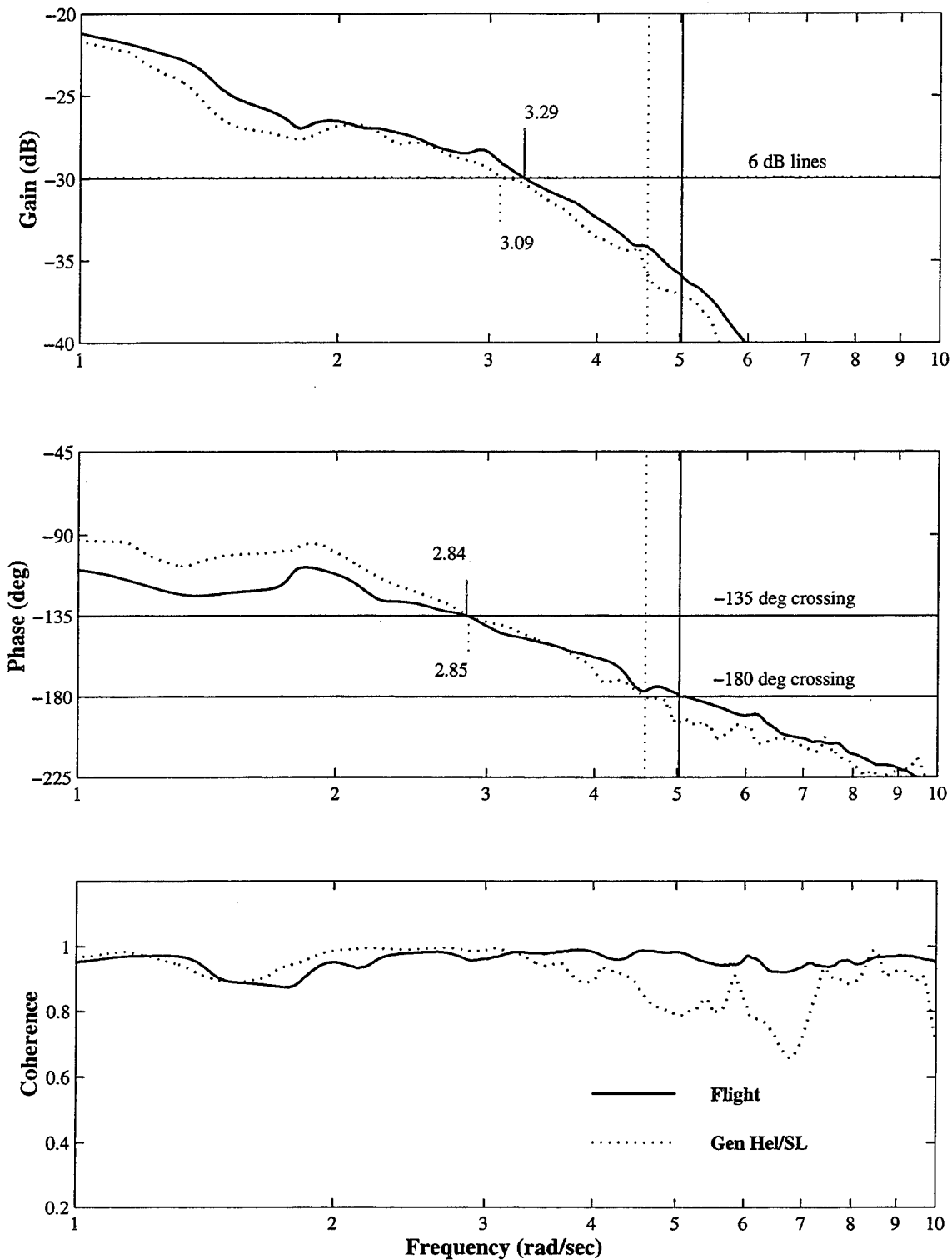


Figure E.4. 4K Block Handling Quality Determination, Longitudinal Axis, (a) Hover

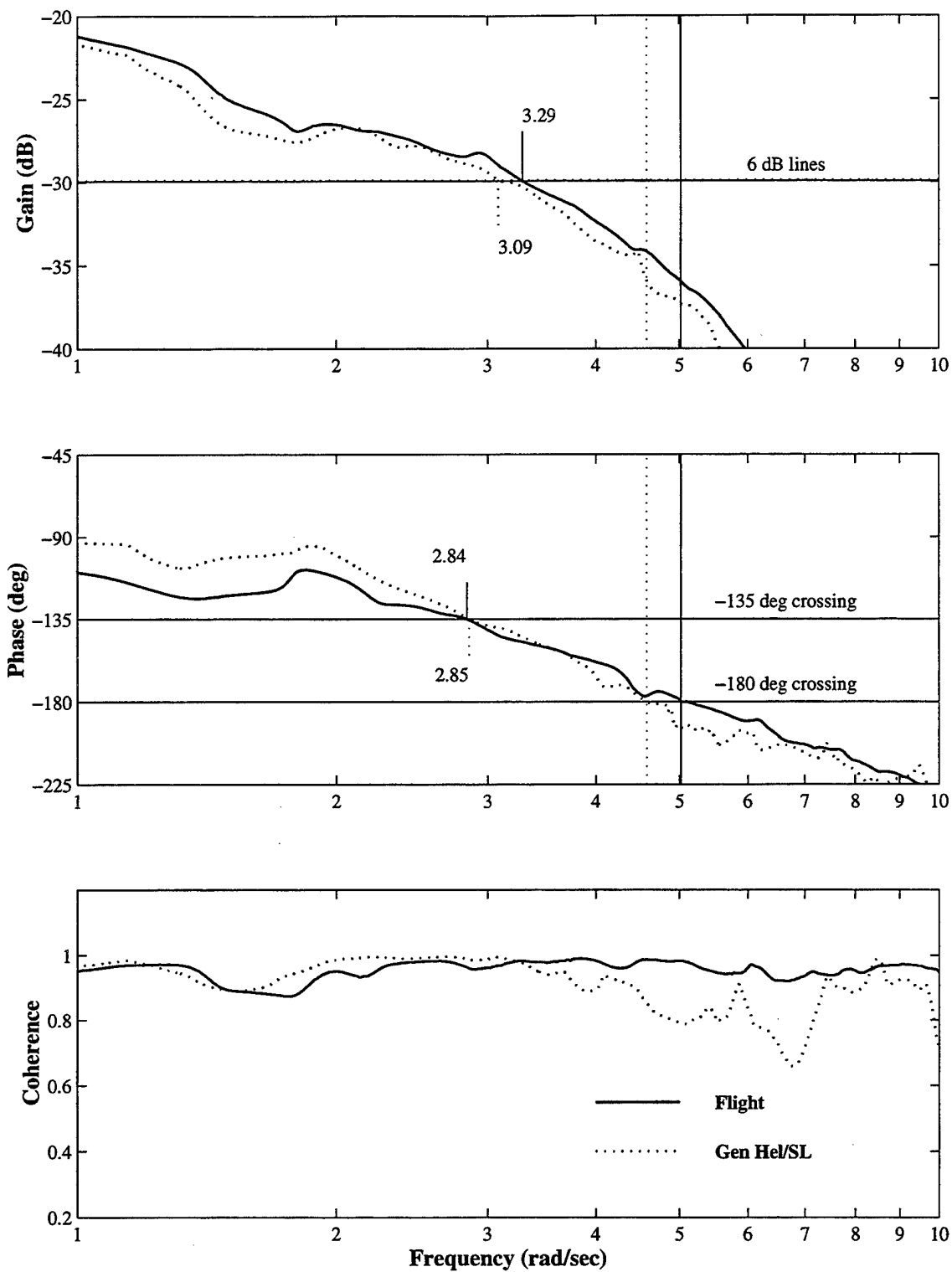


Figure E.4. Continued, (b) 30 Knots

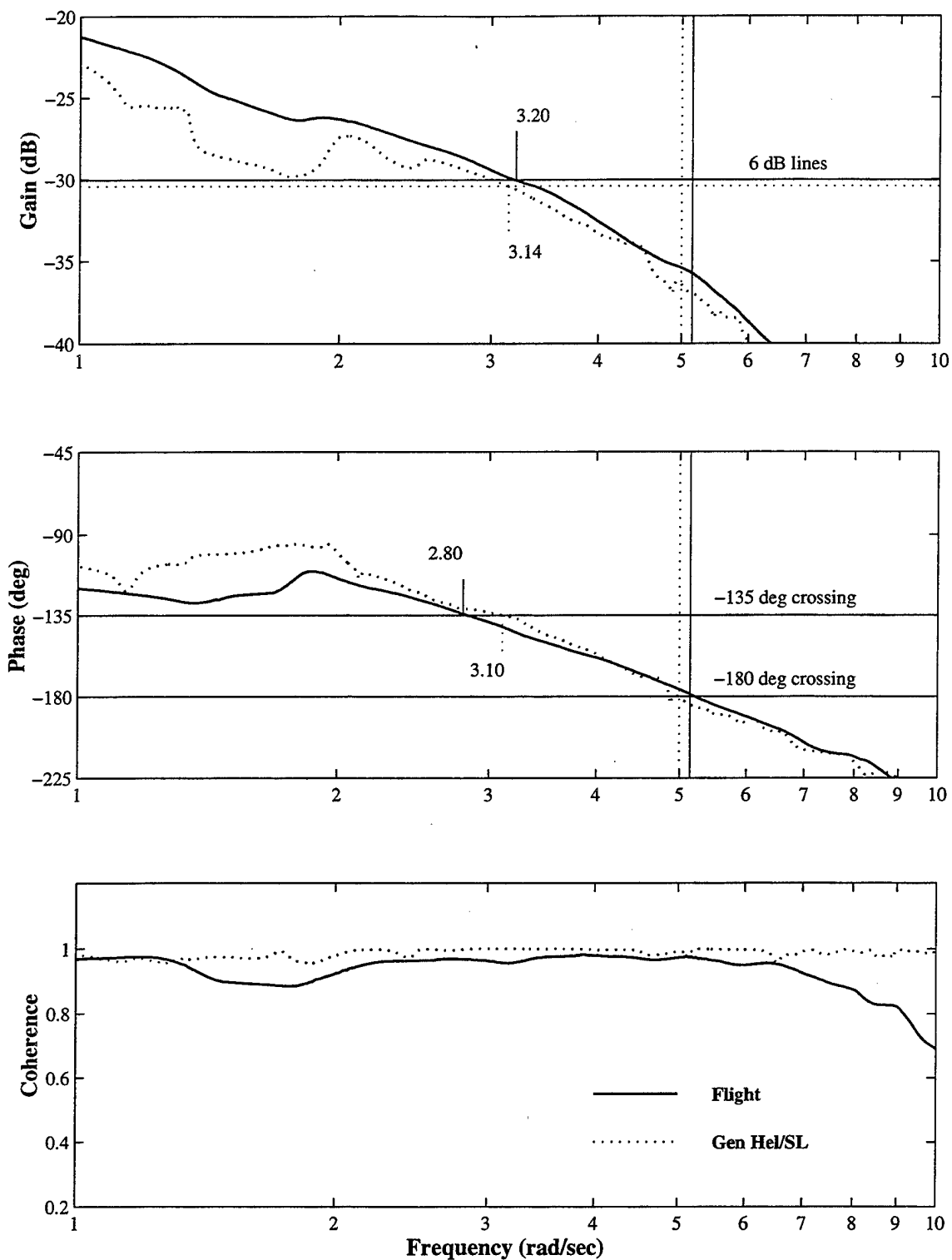


Figure E.4. Continued, (c) 50 Knots

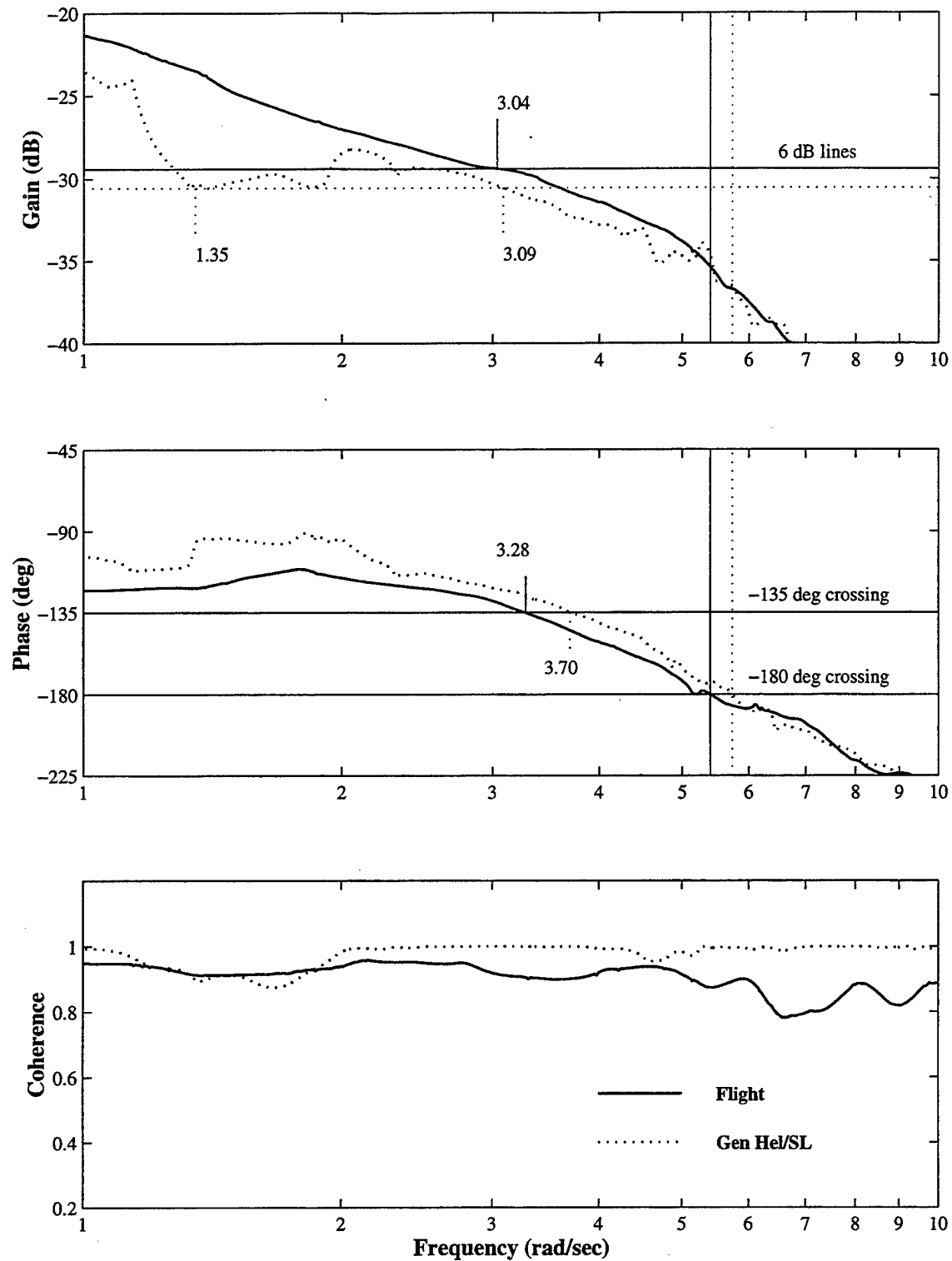


Figure D.4. Continued, (d) 80 Knots



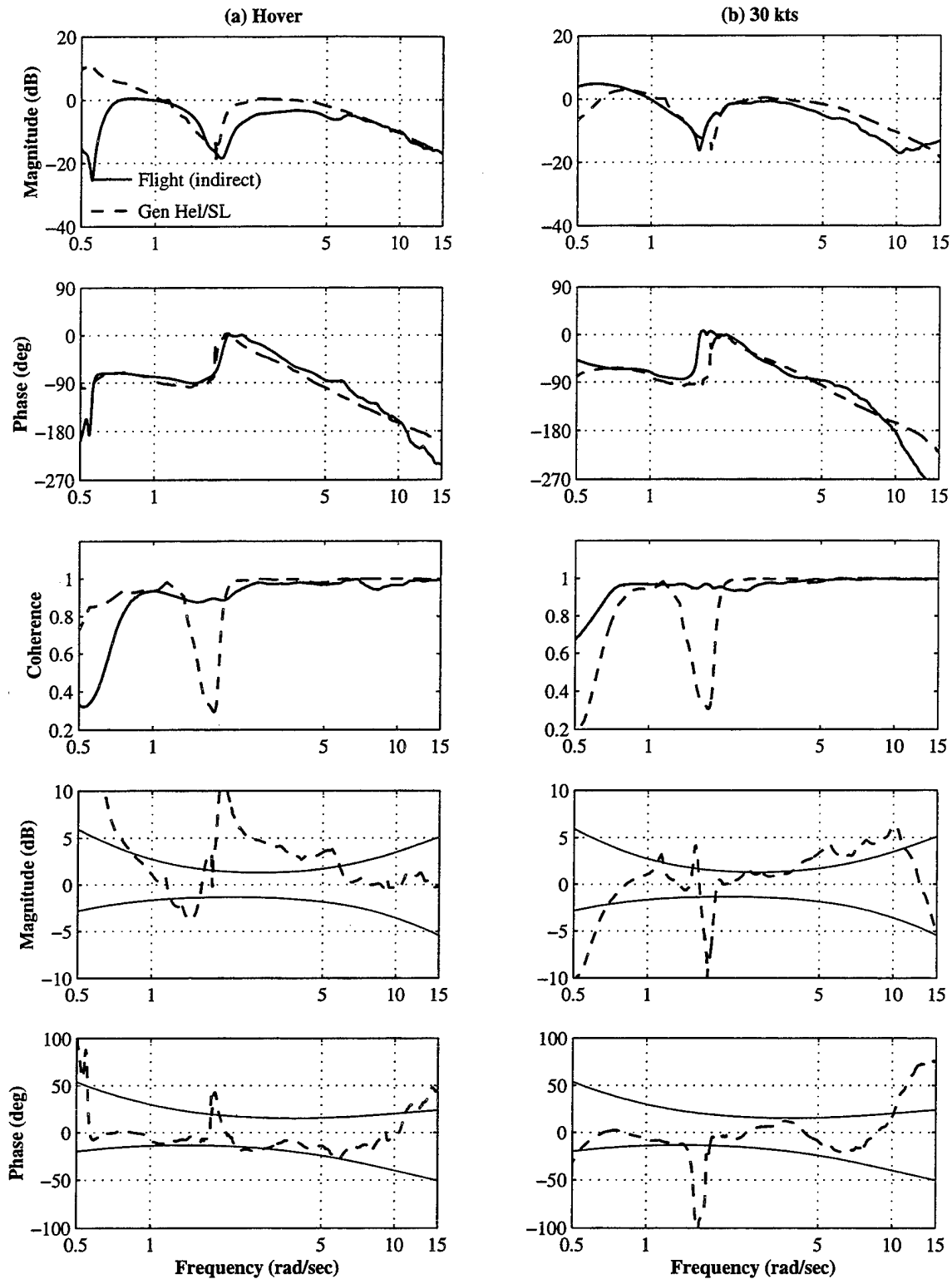


Figure E.5. 4K Block Stability Margin, Lateral Axis, (a) Hover, (b) 30 Knots

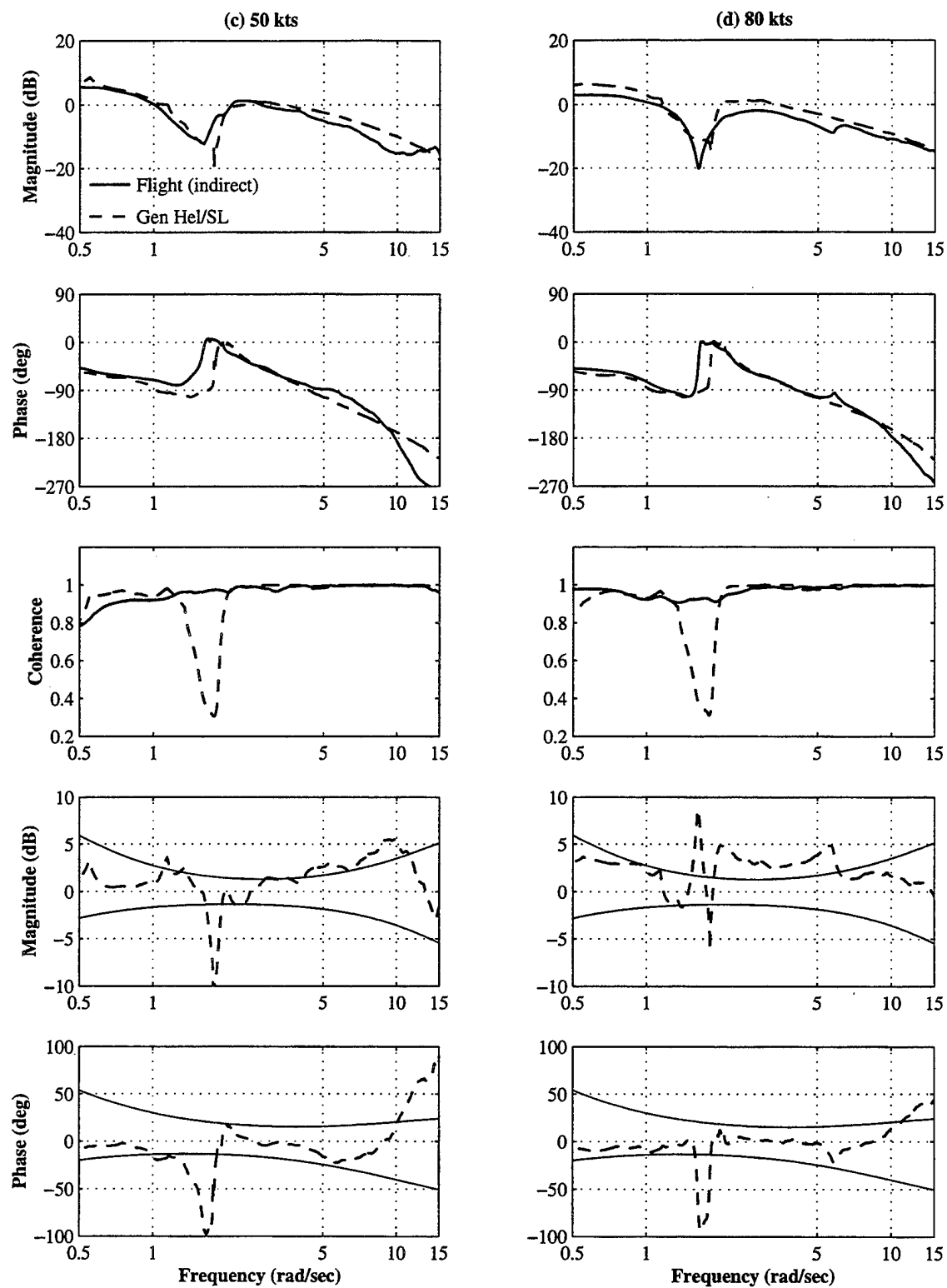


Figure E.5. Continued, (c) 50 Knots, (d) 80 Knots

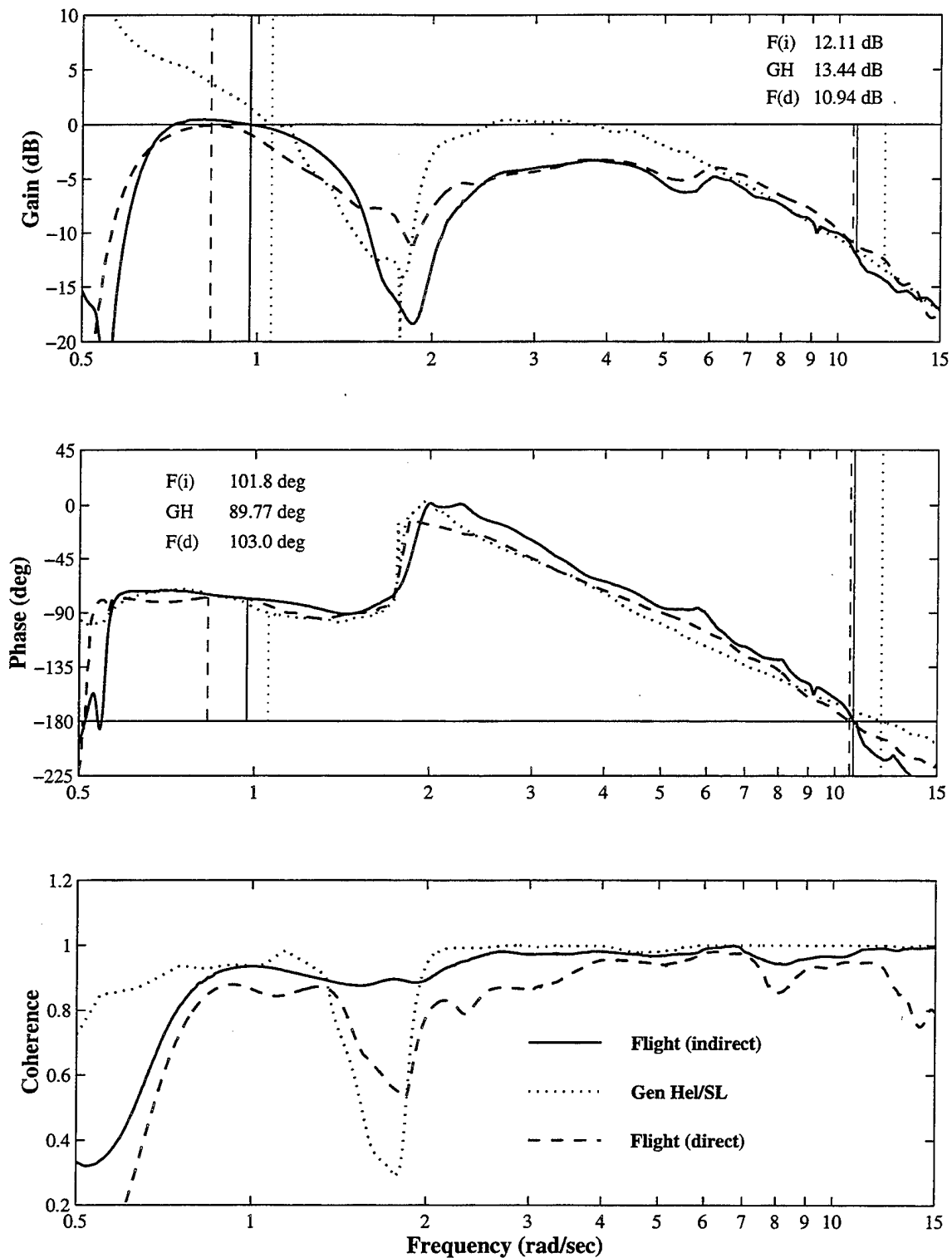


Figure E.6. 4K Block Stability Margin Determination, Lateral Axis, (a) Hover

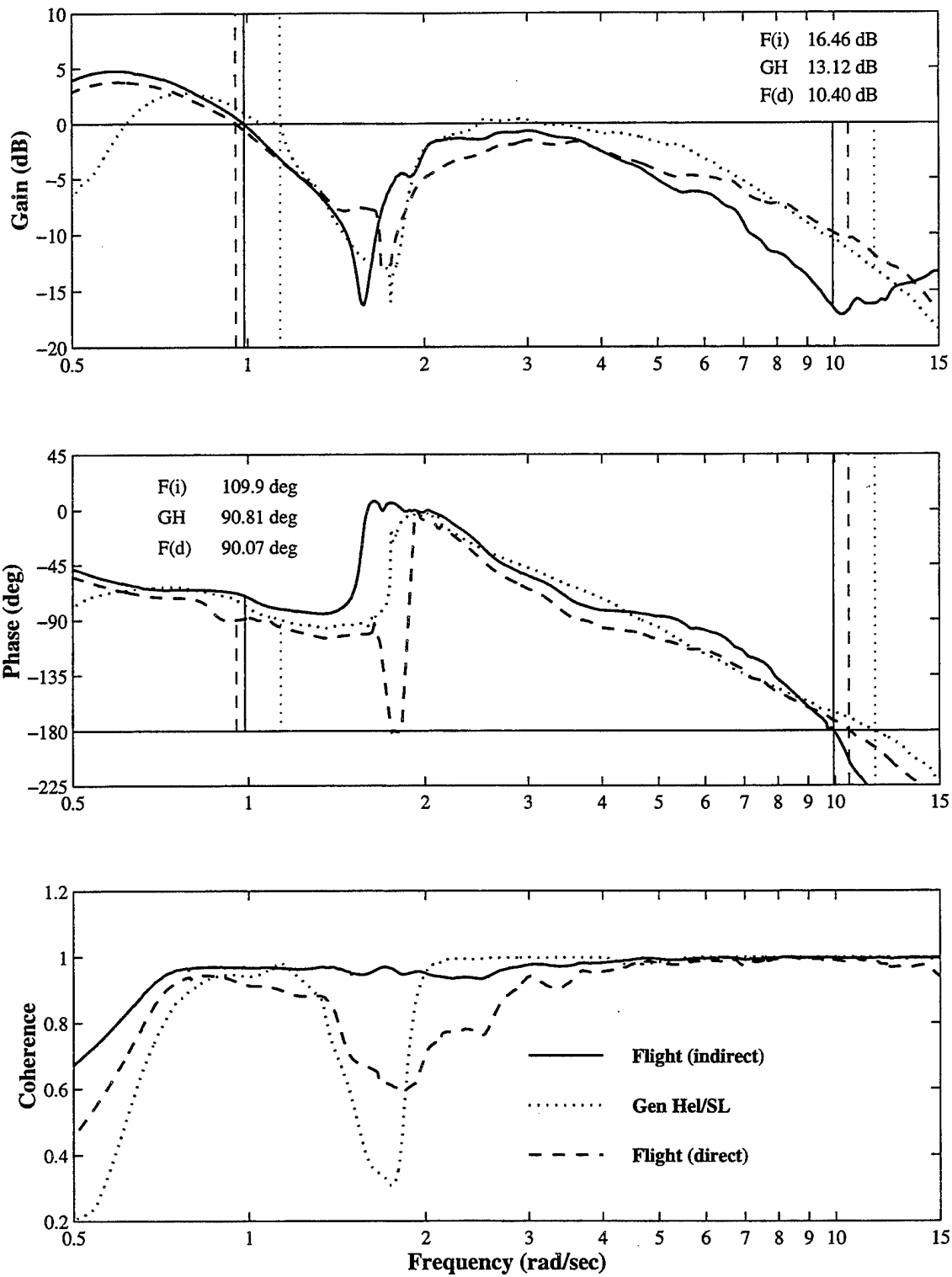


Figure E.6. Continued, (b) 30 Knots

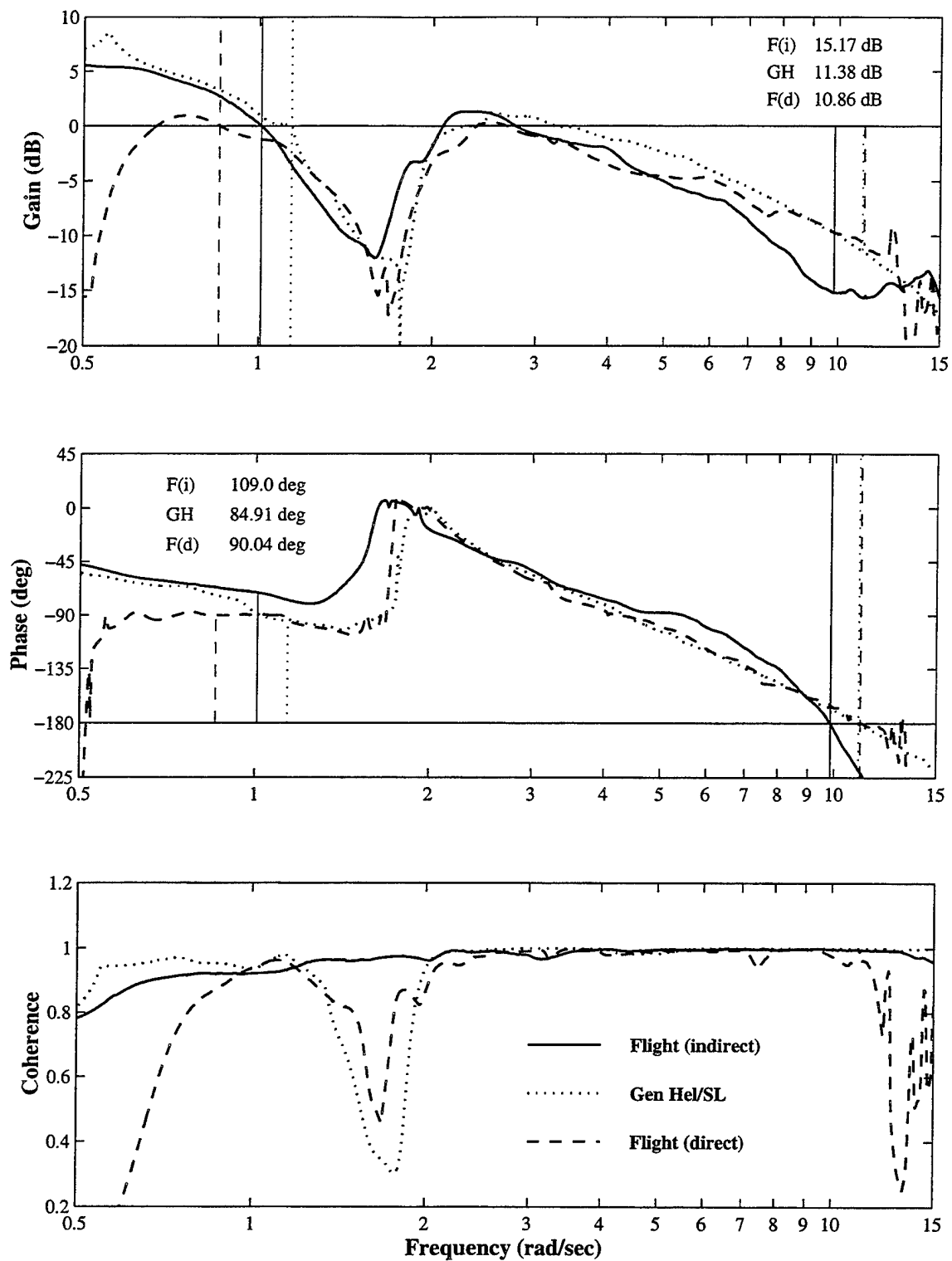


Figure E.6. Continued, (c) 50 Knots

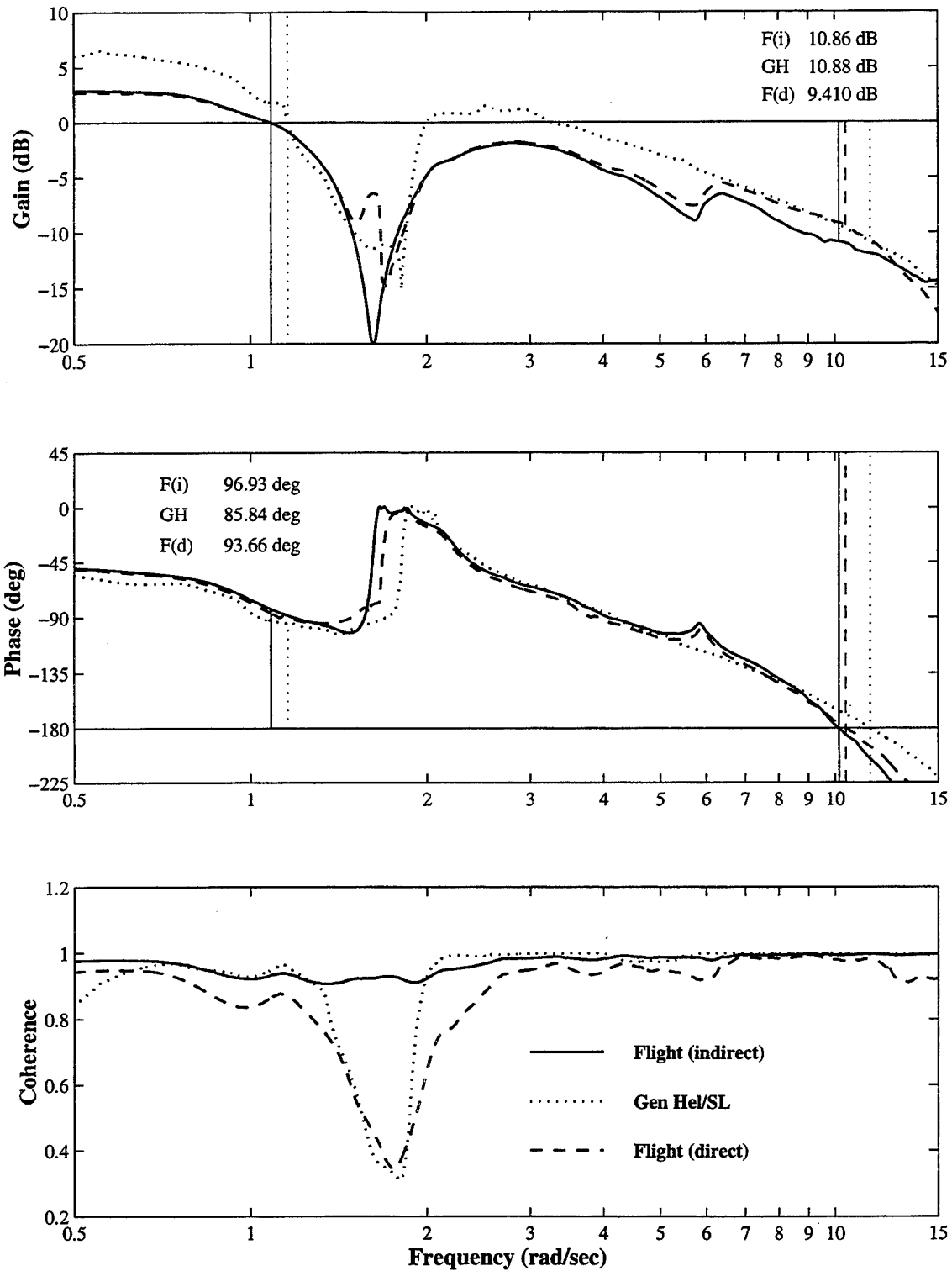


Figure E.6. Continued, (d) 80 Knots

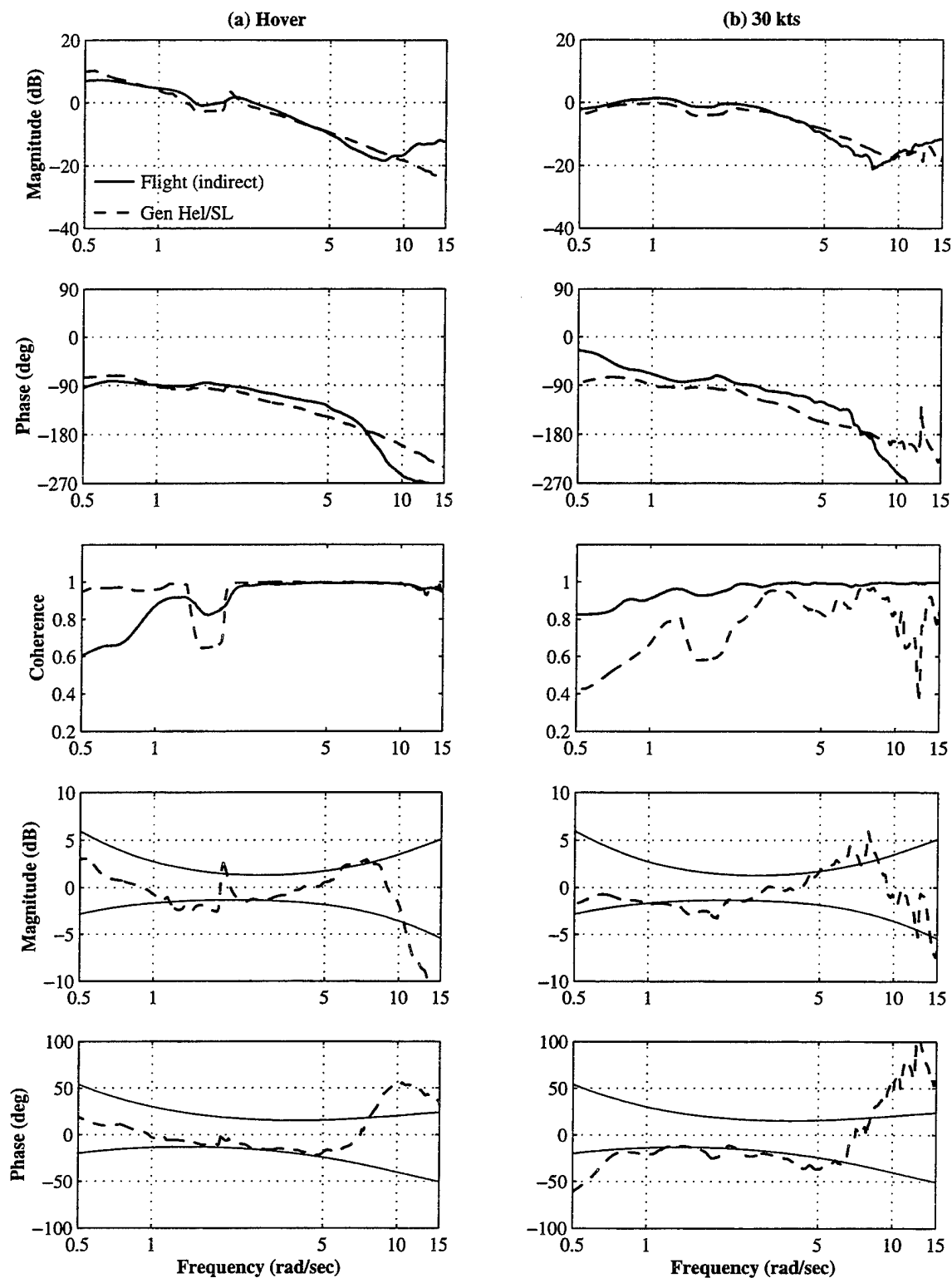


Figure E.7. 4K Block Stability Margin, Longitudinal Axis, (a) Hover, (b) 30 Knots

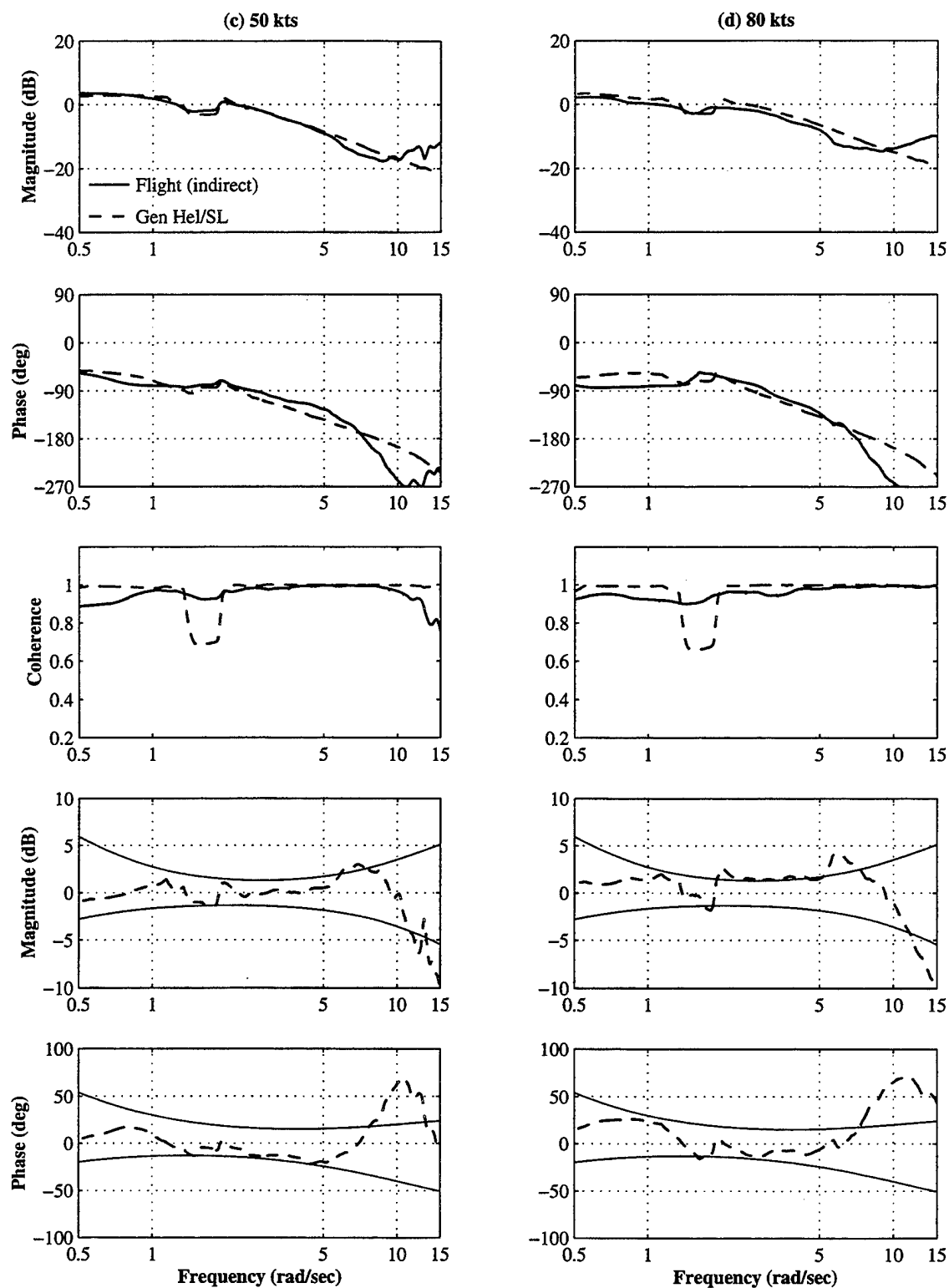


Figure E.7. Continued, (c) 50 Knots, (d) 80 Knots



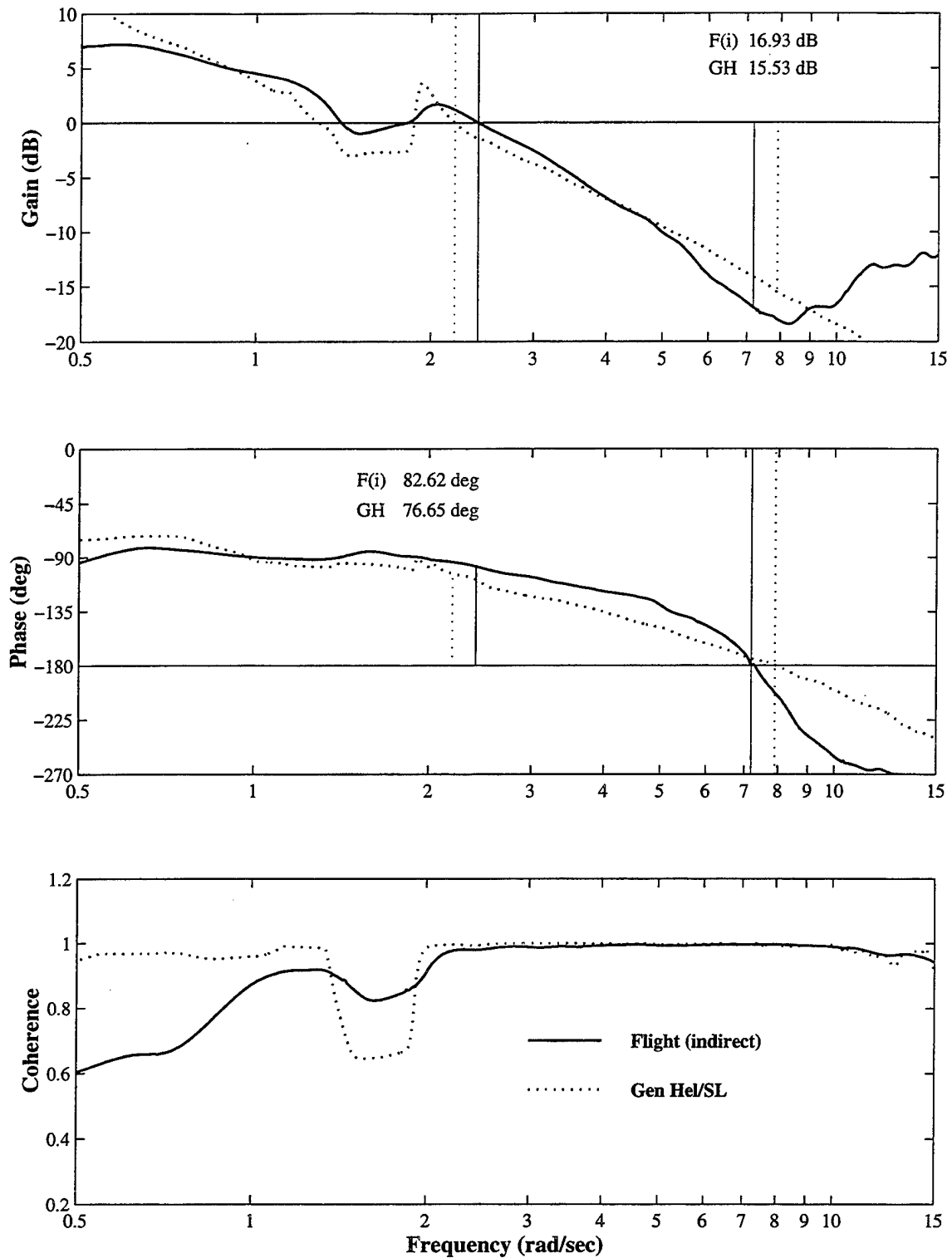


Figure E.8. 4K Block Stability Margin Determination, Longitudinal Axis, (a) Hover

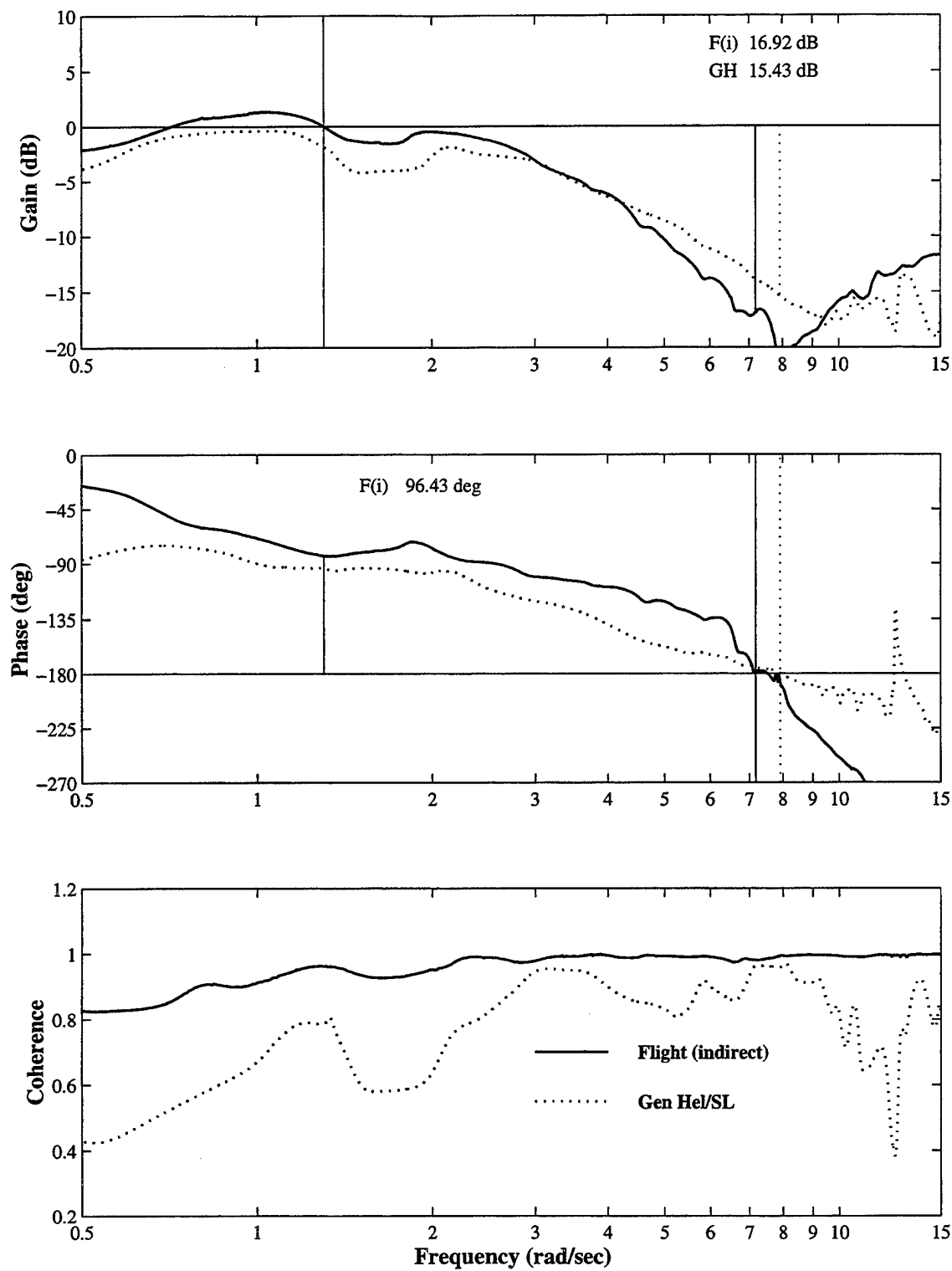


Figure E.8. Continued, (b) 30 Knots

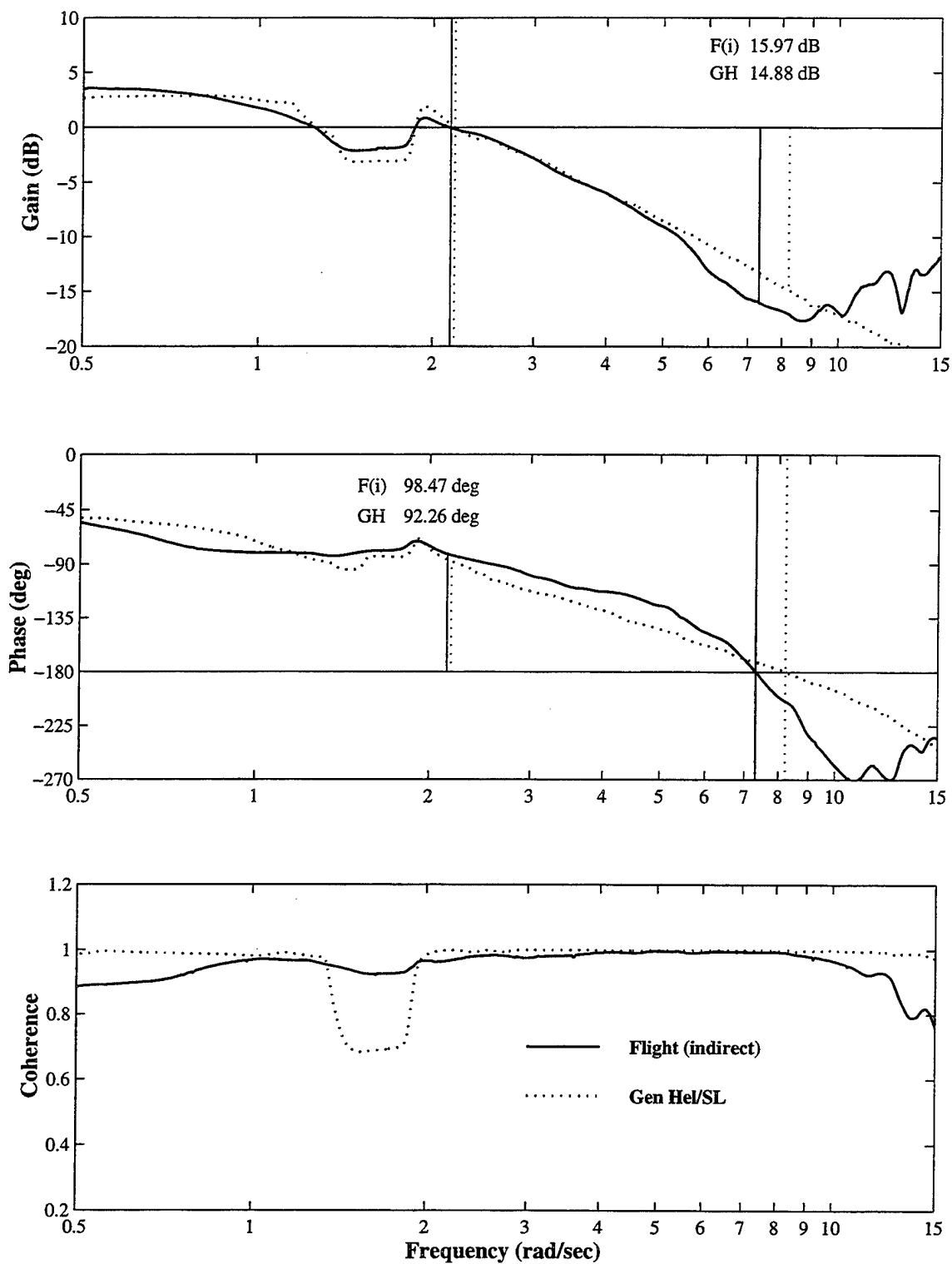


Figure E.8. Continued, (c) 50 Knots

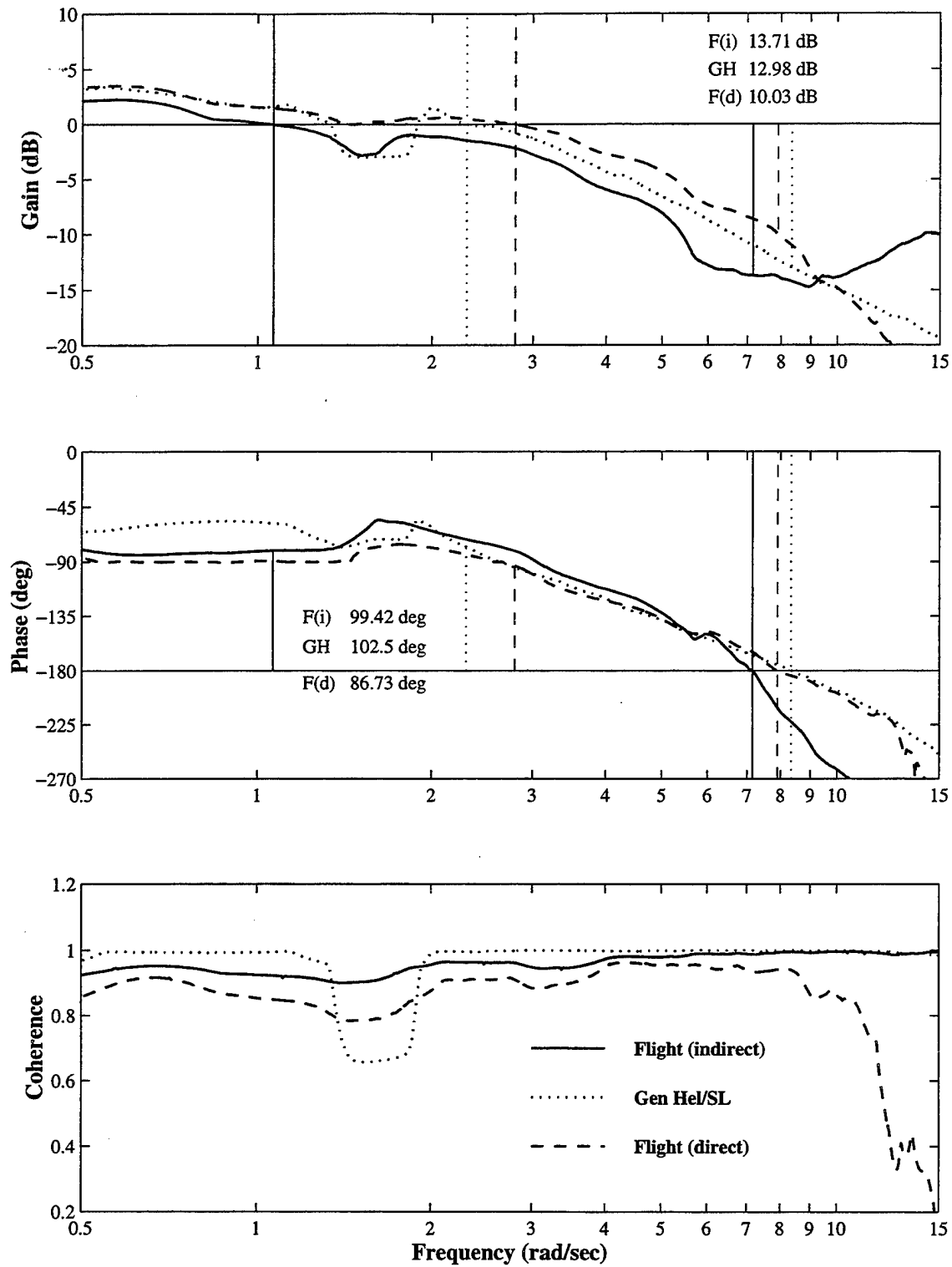


Figure E.8. Continued, (d) 80 Knots

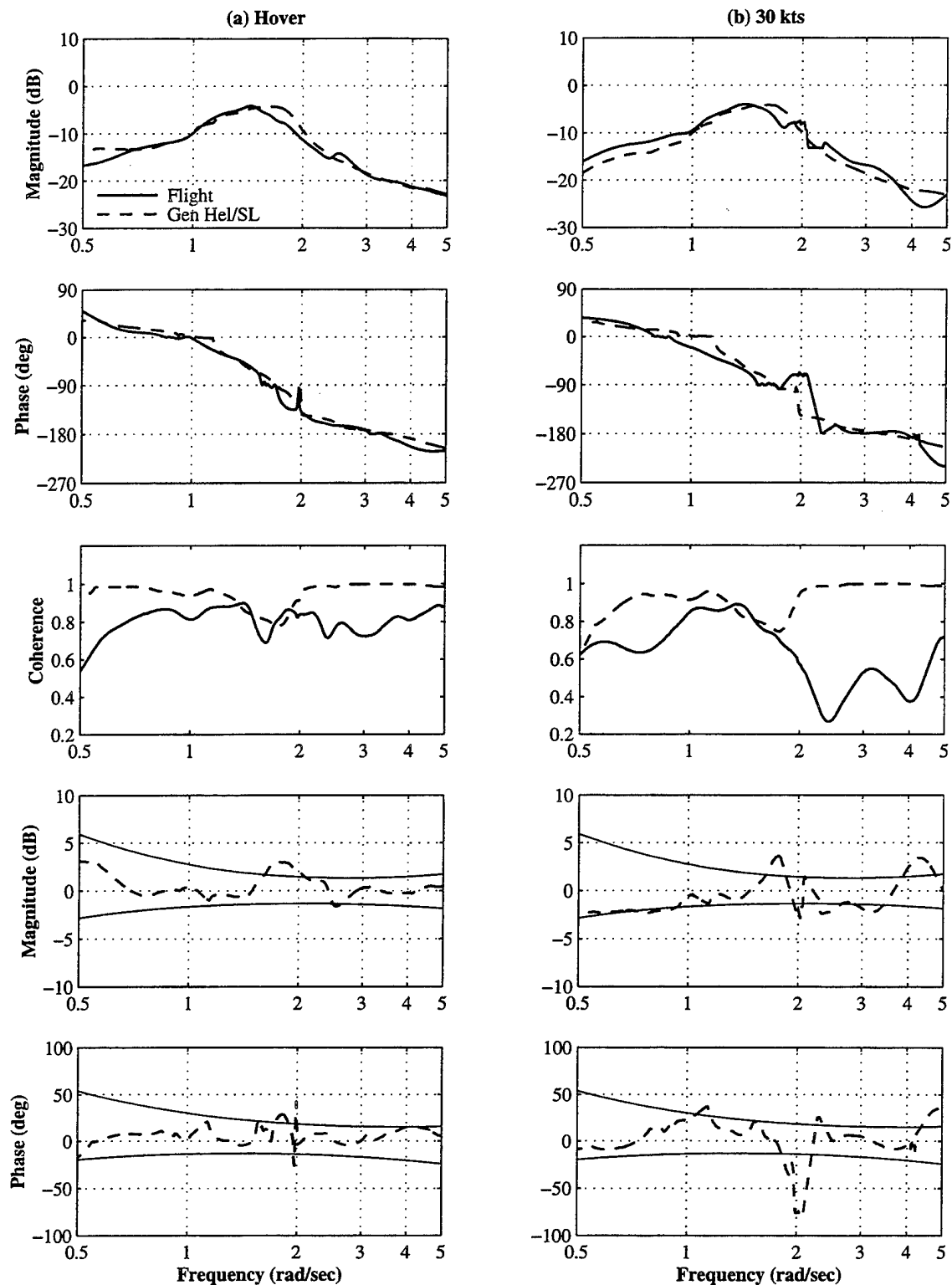


Figure E.9. 4K Block Load Motion, Lateral Axis, (a) Hover, (b) 30 Knots

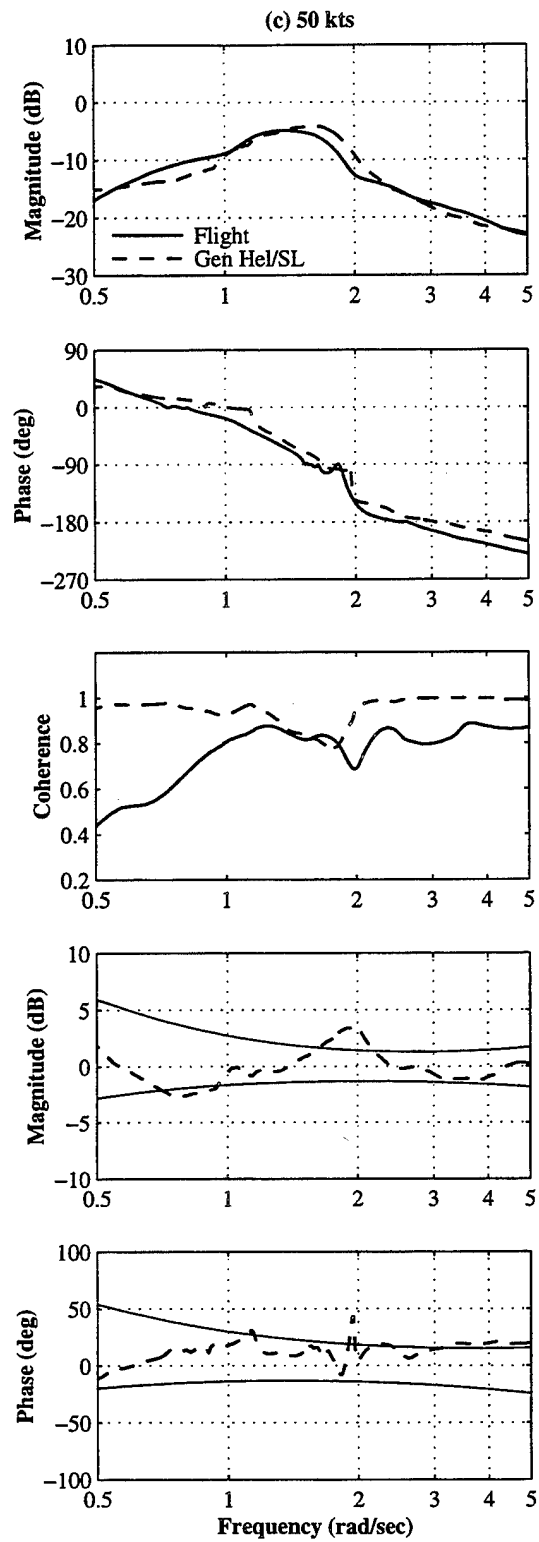


Figure E.9. Continued, (c) 50 Knots

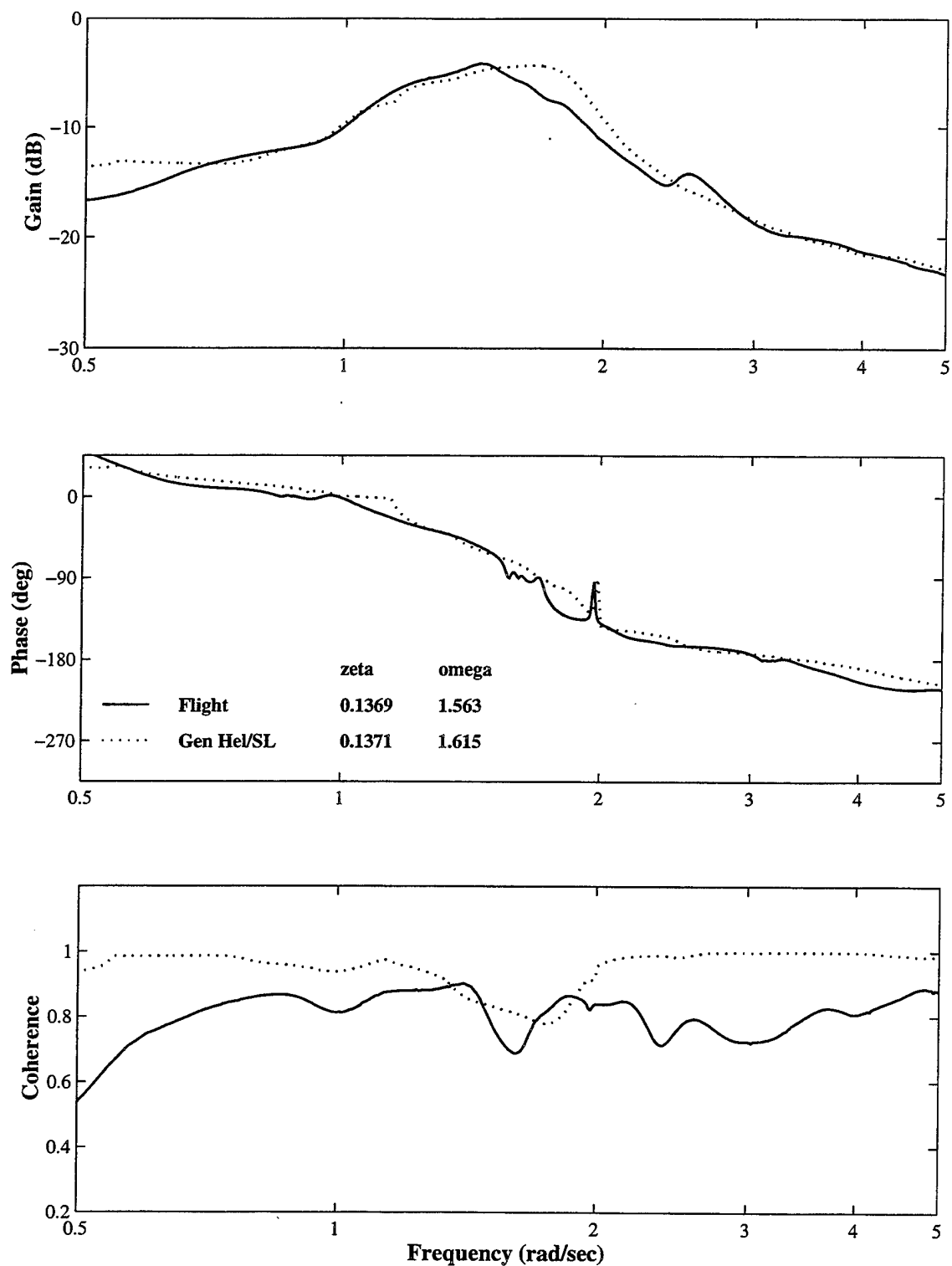


Figure E.10. 4K Block Load Motion Determination, Lateral Axis, (a) Hover

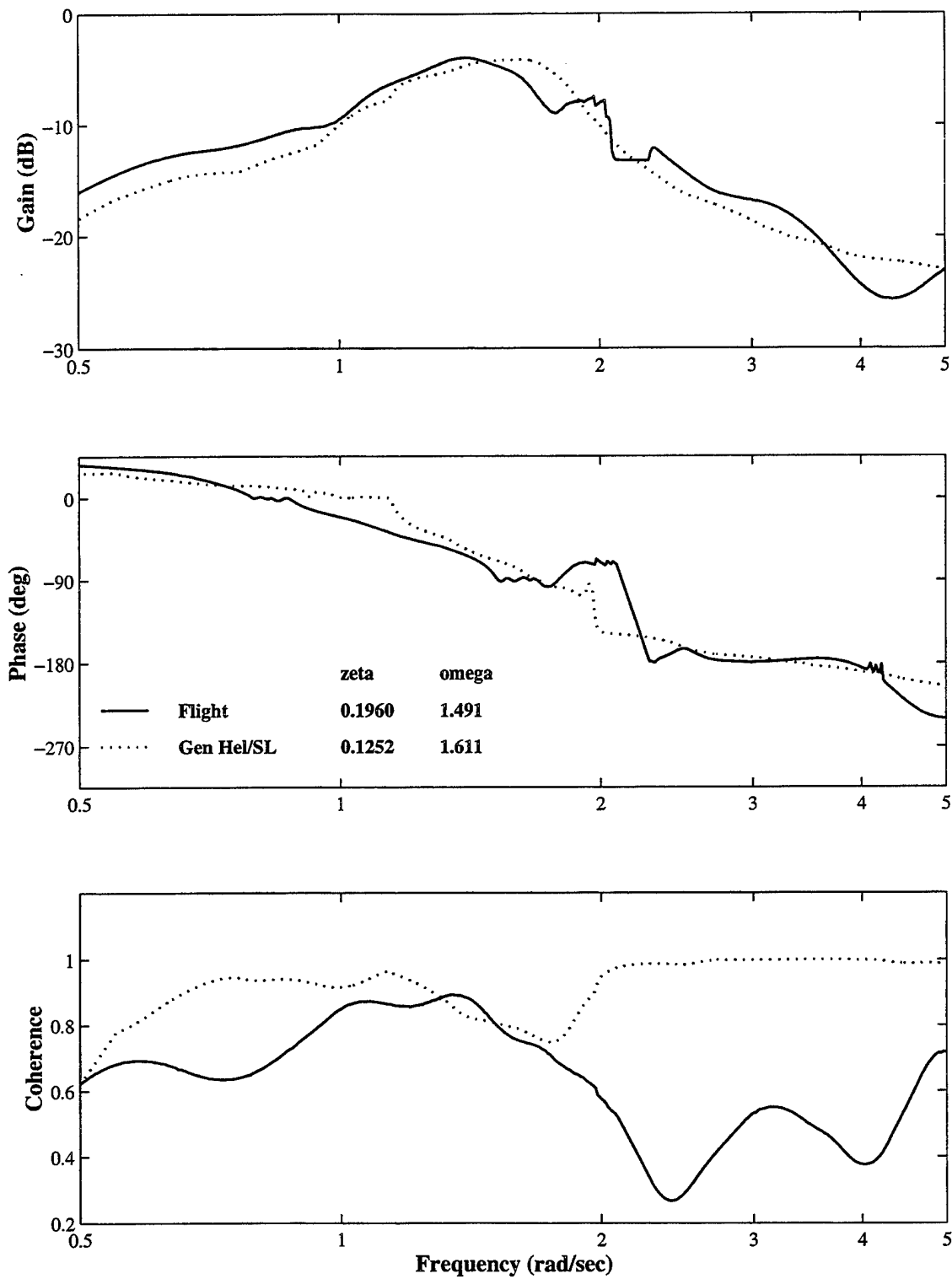


Figure E.10. Continued, (b) 30 Knots



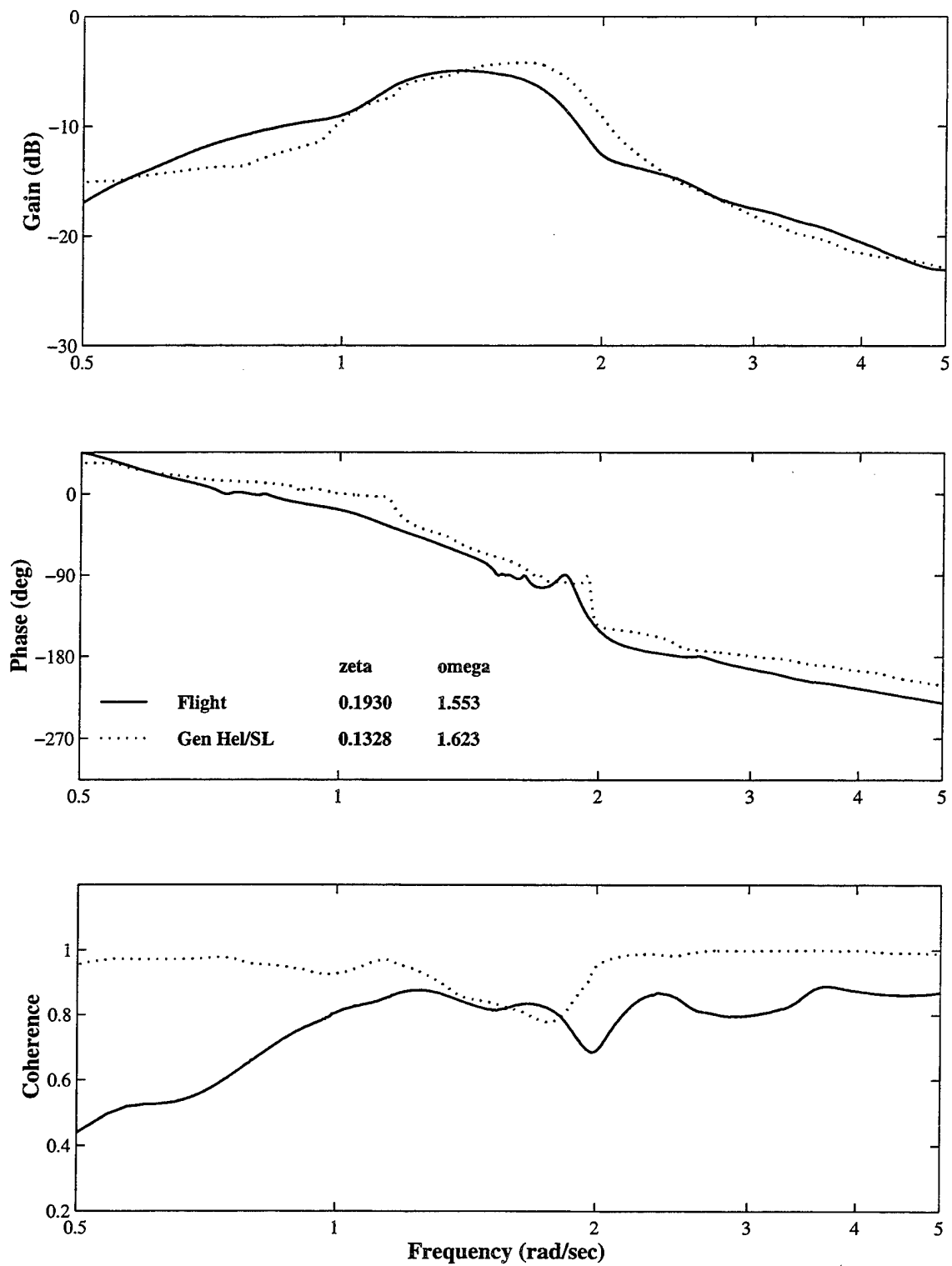


Figure E.10. Continued, (c) 50 Knots

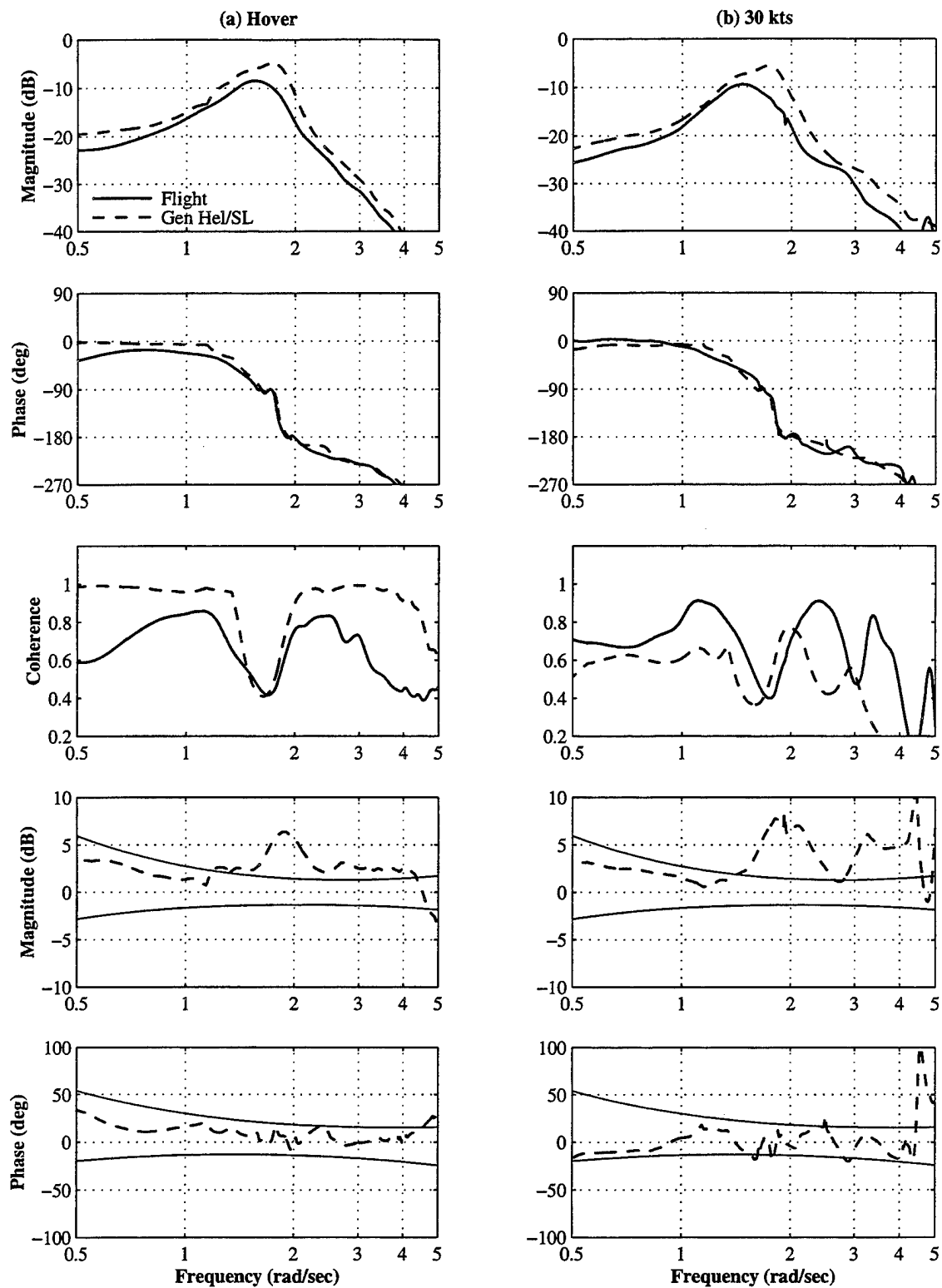


Figure E.11. 4K Block Load Motion, Longitudinal Axis, (a) Hover, (b) 30 Knots

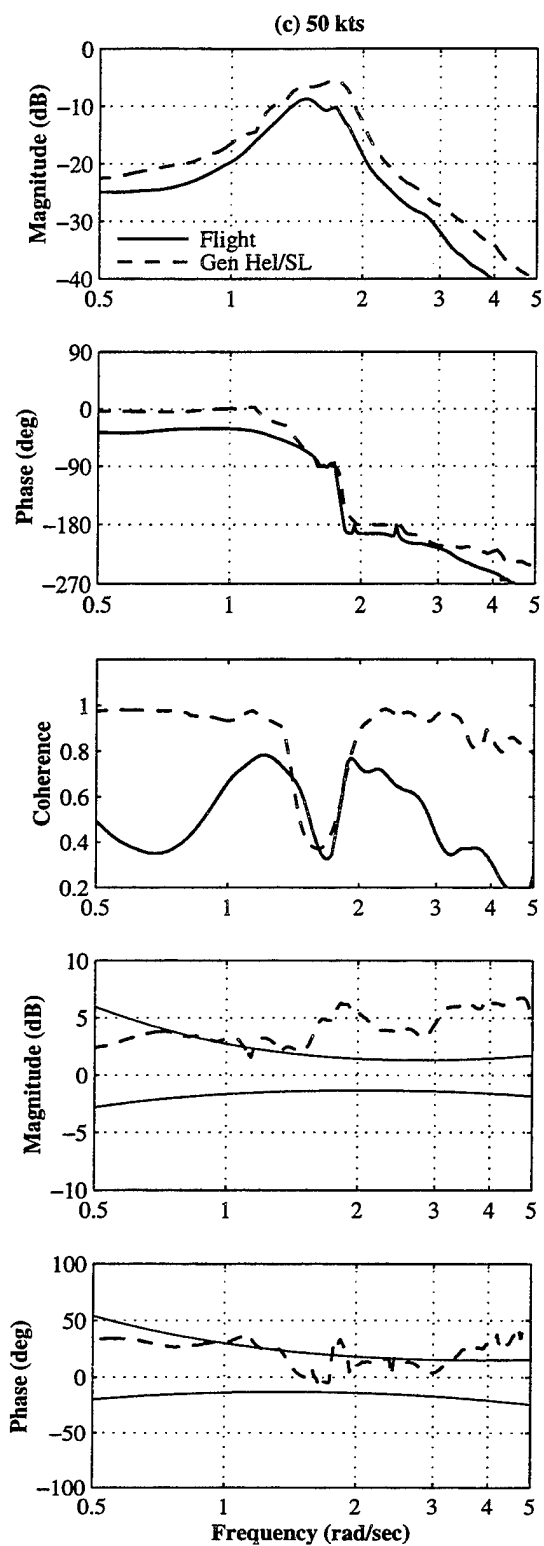


Figure E.11. Continued, (c) 50 Knots

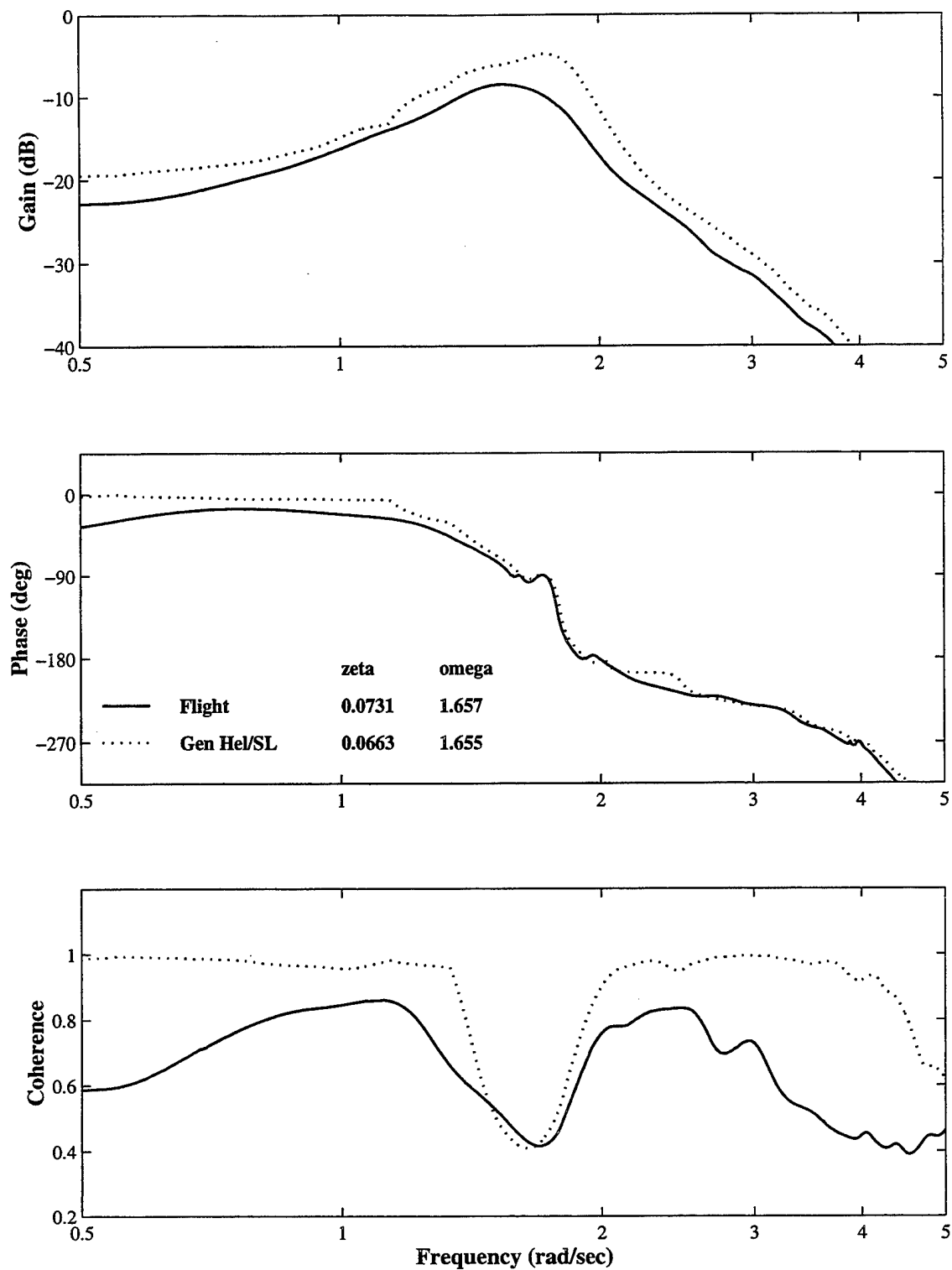


Figure E.12. 4K Block Load Motion Determination, Longitudinal Axis, (a) Hover

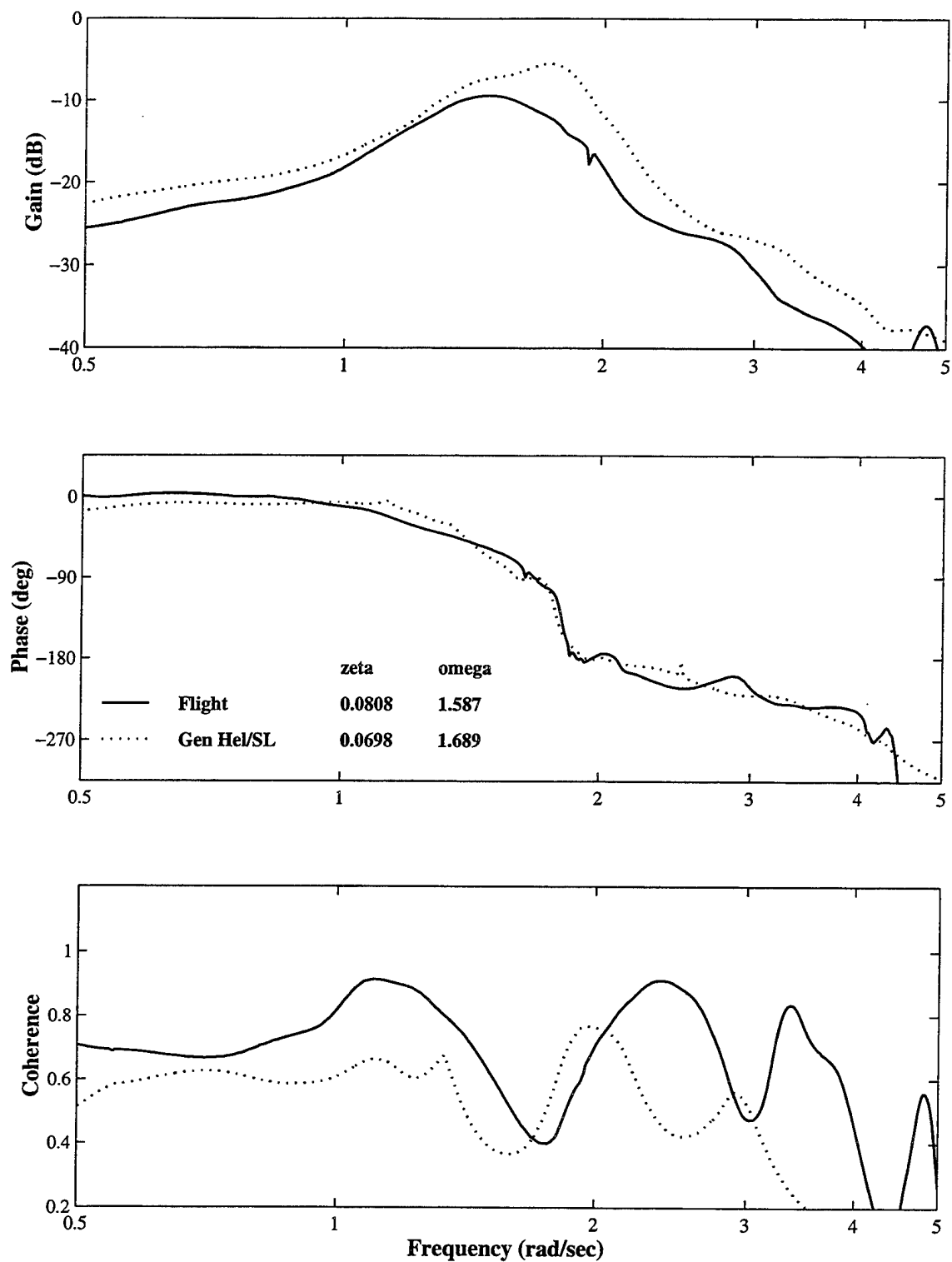


Figure E.12. Continued, (b) 30 Knots

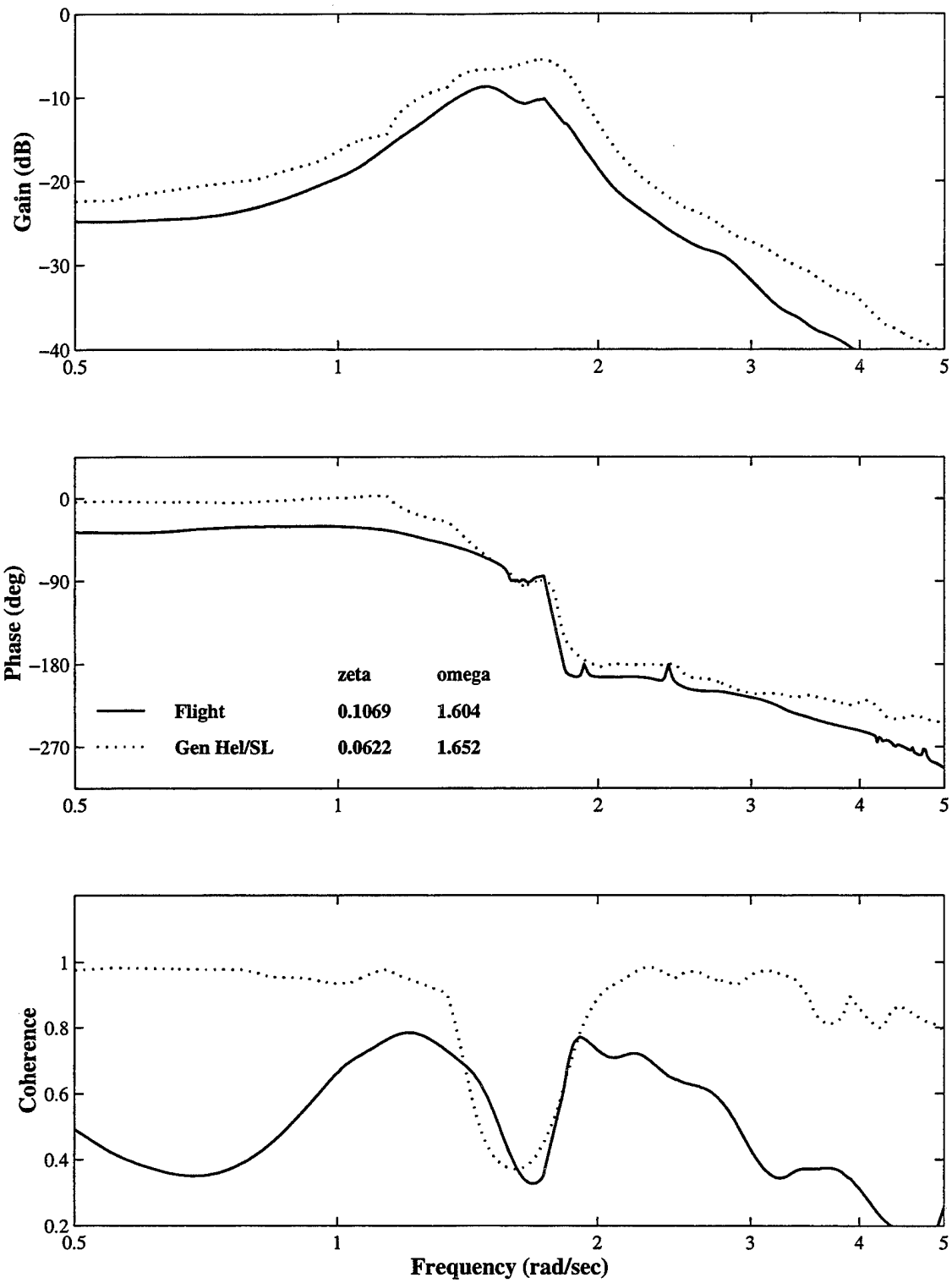


Figure E.12. Continued, (c) 50 Knots



## APPENDIX F. 4K CONEX LOAD DATA

This Appendix contains a summary of the data found for the 4K CONEX configuration from flight data as compared to Gen Hel/SL. In Table F.1, the cases presented at each airspeed for the lateral and longitudinal axes are given as (all Gen Hel/SL responses for HQ and SM are corrected with the average gain and time delay factors):

<b>Fi</b>	Flight, indirect method for computing Stability Margins
<b>Fd</b>	Flight, direct method for computing Stability Margins
<b>N</b>	Gen Hel, no load aerodynamic forces and moments
<b>D</b>	Gen Hel, drag only load aerodynamics estimation
<b>C</b>	Gen Hel, CONEX static load aerodynamics

Table F.2 shows the load motion characteristics as computed by NAVFIT over a frequency range of [0.5, 2.5] rad/sec.

Figures F.1 through F.4 show the helicopter attitude response to aircraft attitude as required by HQ analysis. Figures F.1 and F.3 present the overall frequency response across the range [0.5, 20] rad/sec and include magnitude and phase error functions with the proposed Level D criteria. Figures F.2 and F.4 give the same responses over the range [1, 10] rad/sec, providing a detailed view of the HQ parameter determination along with data coherence.

Figures F.5 through F.8 show the broken loop response used in the determination of SM from Flight, measured by both direct and indirect methods, and from the corrected Gen Hel/SL simulation results. Figures F.5 and F.7 provide the data for the range [0.5, 15] rad/sec, and include the error functions. Figures F.6 and F.8 provide details on the SM determination.

Figures F.9 and F.11 present the load motion frequency responses from the Flight data and from the Gen Hel/SL simulation. The Gen Hel/SL data do not have any correction factor applied, and represent no load aerodynamics. The frequency responses, coherence, and error functions are shown over the range [0.5 5] rad/sec.



Case		-180° Crossing (rad/s)	-180° Gain (dB)	-135° Band- width (rad/s)	6 dB Band- width (rad/s)	Phase Delay (s)	LS Fit Range (rad/s)	Gain Margin (dB)	Gain Crossing (rad/s)	Phase Margin (deg)	Phase Crossing (rad/s)
L A T E R A L	0	Fi	8.038	-30.37	6.216	0.1634	2-8	21.00	7.133	86.37	2.433
		Fd						11.67	10.38	-	-
		N	7.681	-32.72	5.385	0.1368		13.28	12.09	158.7 112.5	2.254 3.707
		D	7.746	-32.77	5.366	0.1403		13.22	12.04	111.7	3.744
		C	8.331	-32.81	5.778	0.1438		13.31	12.05	113.3	3.647
	3 0	Fi	8.077	-31.56	5.943	0.1455	2-8	20.38	7.018	105.9 119.9 106.6	1.223 1.618 2.198
		Fd						10.65	10.28	-	-
		N	0.2345	-25.09	0.2097	0.1474		11.97	11.26	151.0 125.6	2.486 3.225
		D	7.662	-32.32	5.390	0.1484		12.16	11.31	125.8	3.201
		C	8.343	-32.44	5.839	0.1525		13.42	11.69	126.0	3.126
	5 0	Fi	8.256	-31.09	5.656	0.1448	2-8	23.60	7.524	99.31	2.527
		Fd						10.12	10.23	-	-
		N	7.590	-32.41	5.050	0.1427		11.70	11.04	166.2 115.7	2.008 3.238
		D	7.592	-32.42	5.012	0.1422		11.68	11.04	115.5	3.250
		C	8.184	-32.42	5.434	0.1448		11.62	11.01	115.3	3.243
	6 0	Fi	8.437	-32.07	6.102	0.1334	2-8	16.21	7.225	103.2 121.0 102.2	1.181 1.656 2.200
		Fd						10.77	10.44	-	-
		N	7.643	-32.46	4.978	0.1360		11.42	11.25	169.8 113.0	1.979 3.200
		D	7.659	-32.52	4.970	0.1357		11.22	11.09	111.2	3.325
		C	8.281	-32.72	5.404	0.1289		11.12	11.05	110.3	3.342
	7 0	Fi	8.324	-31.43	5.890	0.1317	2-8	17.11	7.034	104.5 111.1 91.49	1.228 1.453 2.769
		Fd						11.56	10.13	114.8	0.510
		N	7.744	-33.18	4.758	0.1279		11.38	11.25	164.6 111.0	1.998 3.201
		D	7.768	-33.19	4.806	0.1284		11.42	11.29	111.8	3.167
		C	8.476	-33.25	5.425	0.1277		11.28	11.26	111.1	3.177
		D	5.125	-36.69	3.234	0.1978		14.49	8.209	95.20	2.189
		C	5.230	-36.31	3.290	0.1956		14.79	8.201	93.85	2.155

Table F.1. Helicopter Response Summary: 4K CONEX Load Configuration, Lateral Axis

Case			-180° Crossing (rad/s)	-180° Gain (dB)	-135° Band- width (rad/s)	6 dB Band- width (rad/s)	Phase Delay (s)	LS Fit Range (rad/s)	Gain Margin (dB)	Gain Crossing (rad/s)	Phase Margin (deg)	Phase Crossing (rad/s)
LONGITUDINAL	0	Fi	4.580	-34.82	2.675	3.054	0.1948	2-6	14.66	9.728	126.2	0.8000
		Fd							11.71	7.531	72.65	2.506
		N	4.674	-36.09	2.837	3.179	0.1995		15.12	7.889	86.39 90.26 75.76	1.192 1.750 2.234
		D	4.681	-36.02	2.743	3.196	0.2022		15.16	7.903	75.78	2.232
		C	4.709	-35.64	2.944	3.148	0.2051		15.28	7.923	75.66	2.209
	30	Fi	4.950	-35.66	3.051	3.241	0.2045	2-6	14.32	9.974	118.7	0.8176
		Fd							13.83	8.837	86.40 98.06 90.24	1.106 1.751 2.275
		N	5.050	-35.90	2.839	3.304	0.1983		14.41	8.100	92.97 105.5 84.38	1.217 1.803 2.306
		D	4.948	-36.11	3.014	3.261	0.2056		14.55	8.048	84.40	2.289
		C	5.045	-36.09	3.104	3.231	0.2044		15.30	8.055	-	-
	50	Fi	4.831	-35.13	2.795	2.991	0.1914	2-6	12.93	9.843	112.2	0.8348
		Fd							12.26	8.646	80.14 87.08 81.09	1.163 1.498 2.533
		N	5.045	-36.76	3.074	3.157	0.1980		14.46	8.209	100.4 117.3 95.40	1.185 1.756 2.194
		D	5.125	-36.69	3.234	3.141	0.1978		14.49	8.209	95.20	2.189
		C	5.230	-36.31	3.290	3.013	0.1956		14.79	8.201	93.85	2.155
	60	Fi	5.145	-35.00	3.085	2.752	0.1754	2-6	13.50	10.09	114.3 165.1 164.1 150.9	0.8725 1.984 1.999 2.193
		Fd							10.40	7.879	81.42	1.083
		N	5.256	-35.77	3.310	2.791	0.1869		13.84	8.323	102.3 119.0 97.07	1.155 1.749 2.228
		D	5.478	-37.12	3.373	3.333	0.1874		13.84	8.329	96.72	2.234
		C	5.406	-36.10	3.406	2.905	0.1878		13.91	8.279	95.90	2.199
	70	Fi	4.990	-33.53	3.075	2.622	0.2003	2-6	12.59	10.64	112.6	0.7848
		Fd							9.978	7.542	79.41 84.95 76.63	1.154 1.823 2.555
		N	5.708	-37.14	3.516	3.359	0.1831		13.32	8.263	128.2 102.0	1.762 2.238
		D	5.675	-37.14	3.505	3.293	0.1861		13.32	8.275	101.0	2.253
		C	5.721	-36.59	3.649	3.075	0.1853		13.26	8.214	98.42	2.254

Table F.1. Continued, Longitudinal Axis



Case			Damping Ratio	Natural Frequency	Cost Function
L A T E R A L	0	F	0.1617	1.5218	42.52
		N	0.1587	1.4410	134.5
		D	0.1623	1.4849	108.1
		C	0.1655	1.4595	132.7
	30	F	0.2099	1.4216	70.79
		N	0.1515	1.4562	115.5
		D	0.1606	1.5557	91.27
		C	0.2102	1.5957	54.23
	50	F	0.2045	1.4218	31.16
		N	0.1613	1.4662	139.6
		D	0.1693	1.4851	115.7
		C	0.1942	1.4958	89.30
	60	F	0.1601	1.5933	73.06
		N	0.1579	1.4592	142.1
		D	0.1686	1.4857	114.2
		C	0.2117	1.5065	95.45
	70	F	0.2155	1.4606	18.39
		N	0.1478	1.4947	142.2
		D	0.1566	1.5318	114.9
		C	0.2151	1.5341	108.5
L O N G I T U D I N A L	0	F	0.1108	1.5000	56.80
		N	0.0780	1.4515	67.97
		D	0.0822	1.4951	56.13
		C	0.0829	1.4996	55.62
	30	F	0.1004	1.4393	71.08
		N	0.0805	1.4751	67.70
		D	0.0883	1.5629	59.97
		C	0.0942	1.5962	54.82
	50	F	0.1660	1.4431	20.99
		N	0.0756	1.4771	88.36
		D	0.0921	1.4847	83.89
		C	0.0935	1.4998	72.93
	60	F	0.2030	1.4955	27.42
		N	0.0758	1.4777	87.10
		D	0.0969	1.4874	81.05
		C	0.0986	1.5084	67.35
	70	F	0.0891	1.5157	43.71
		N	0.0769	1.4959	78.32
		D	0.1000	1.5049	75.95
		C	0.1055	1.5440	46.70

Table F.2. Load Motion Parameters: 4K CONEX Load Configuration

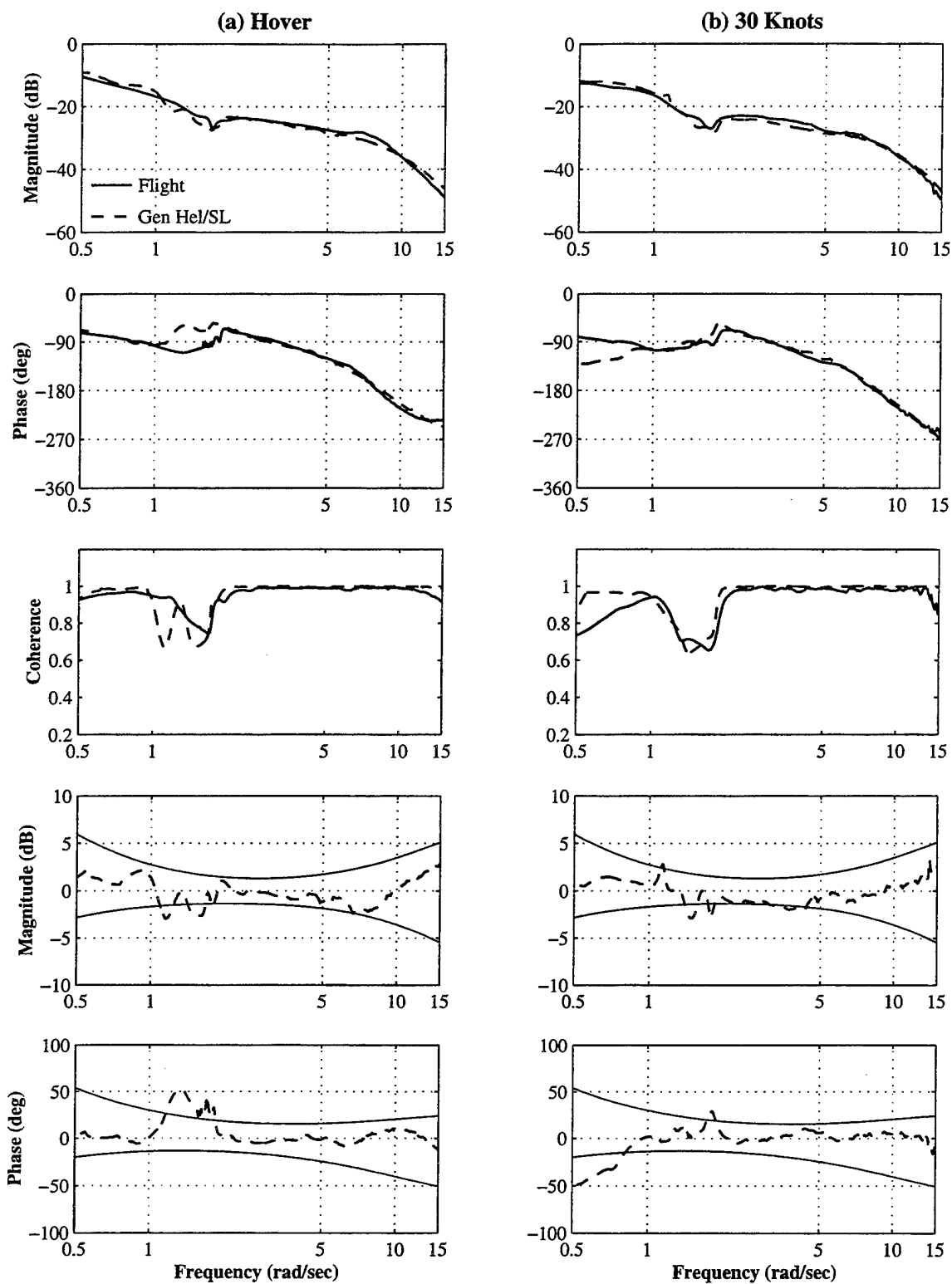


Figure F.1. 4K CONEX Handling Quality, Lateral Axis, (a) Hover, (b) 30 Knots

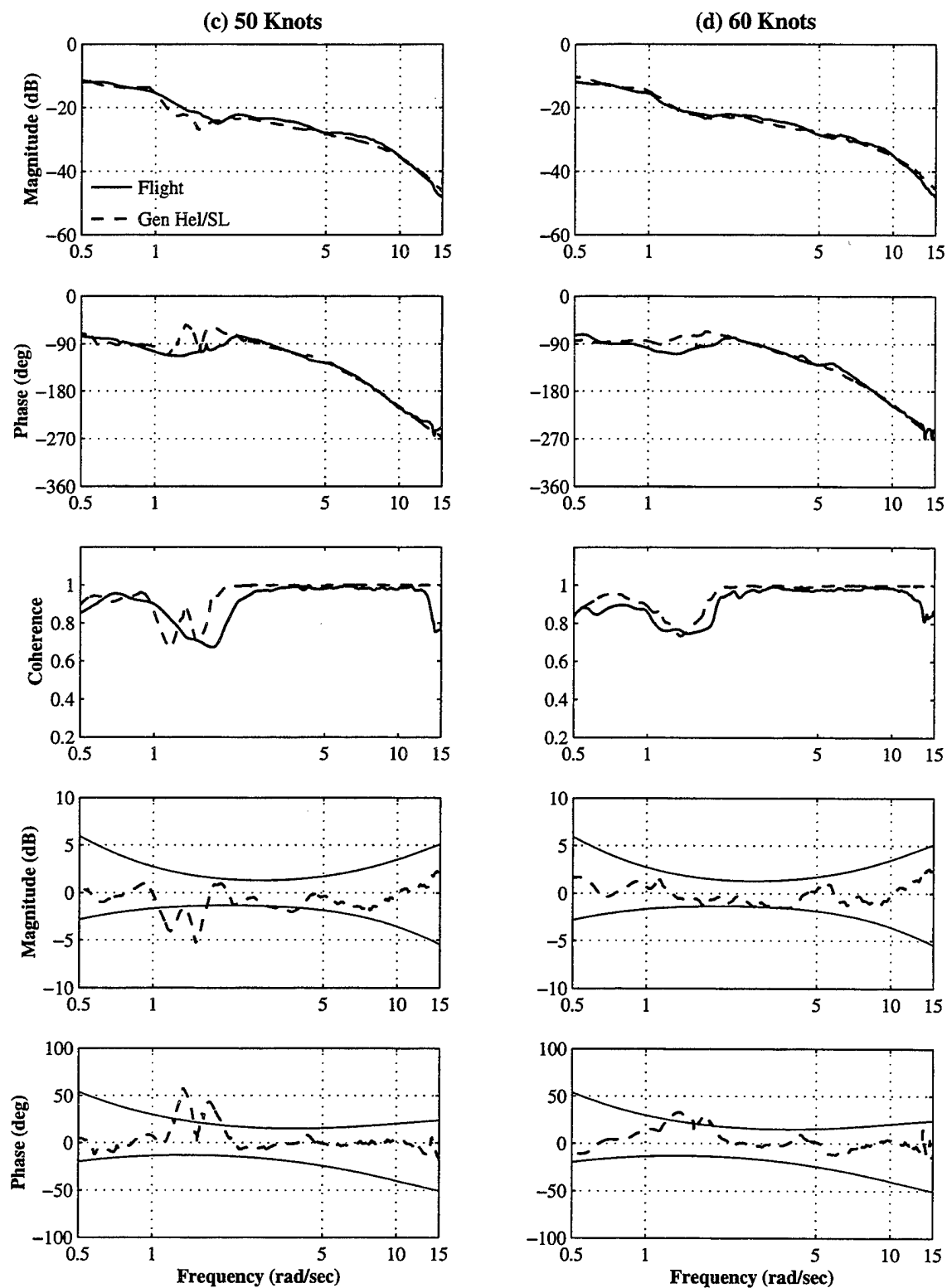


Figure F.1. Continued, (c) 50 Knots, (d) 60 Knots

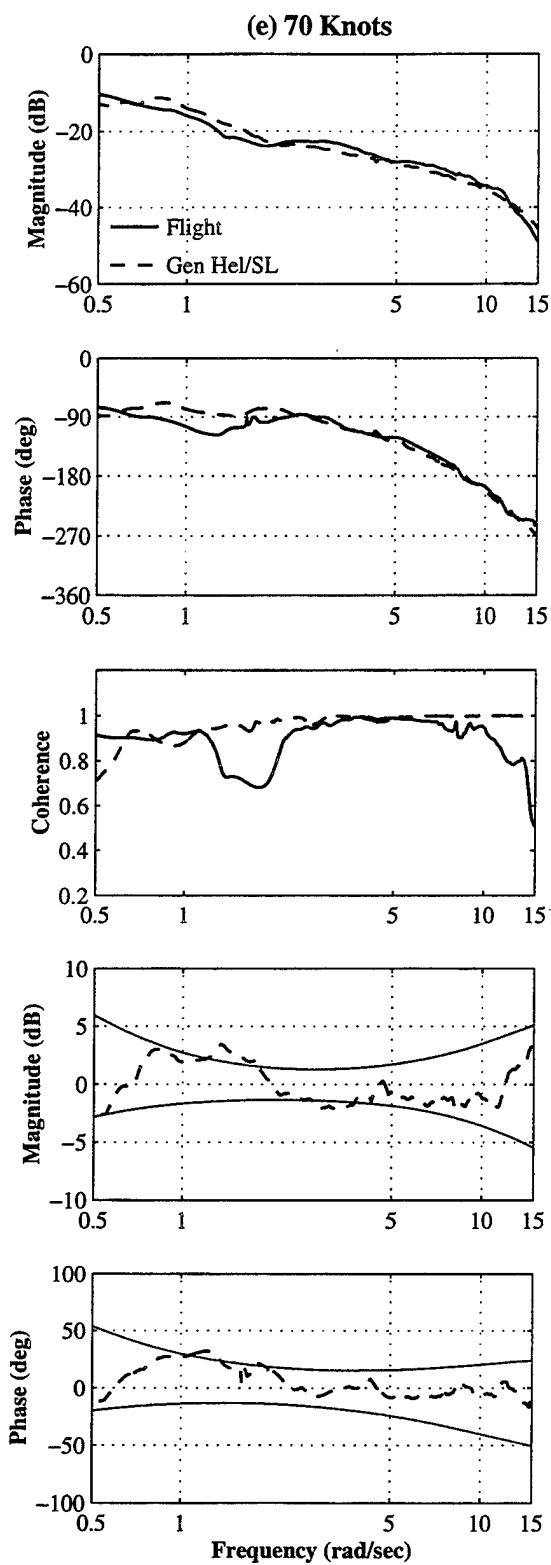


Figure F.1. Continued, (e) 70 Knots

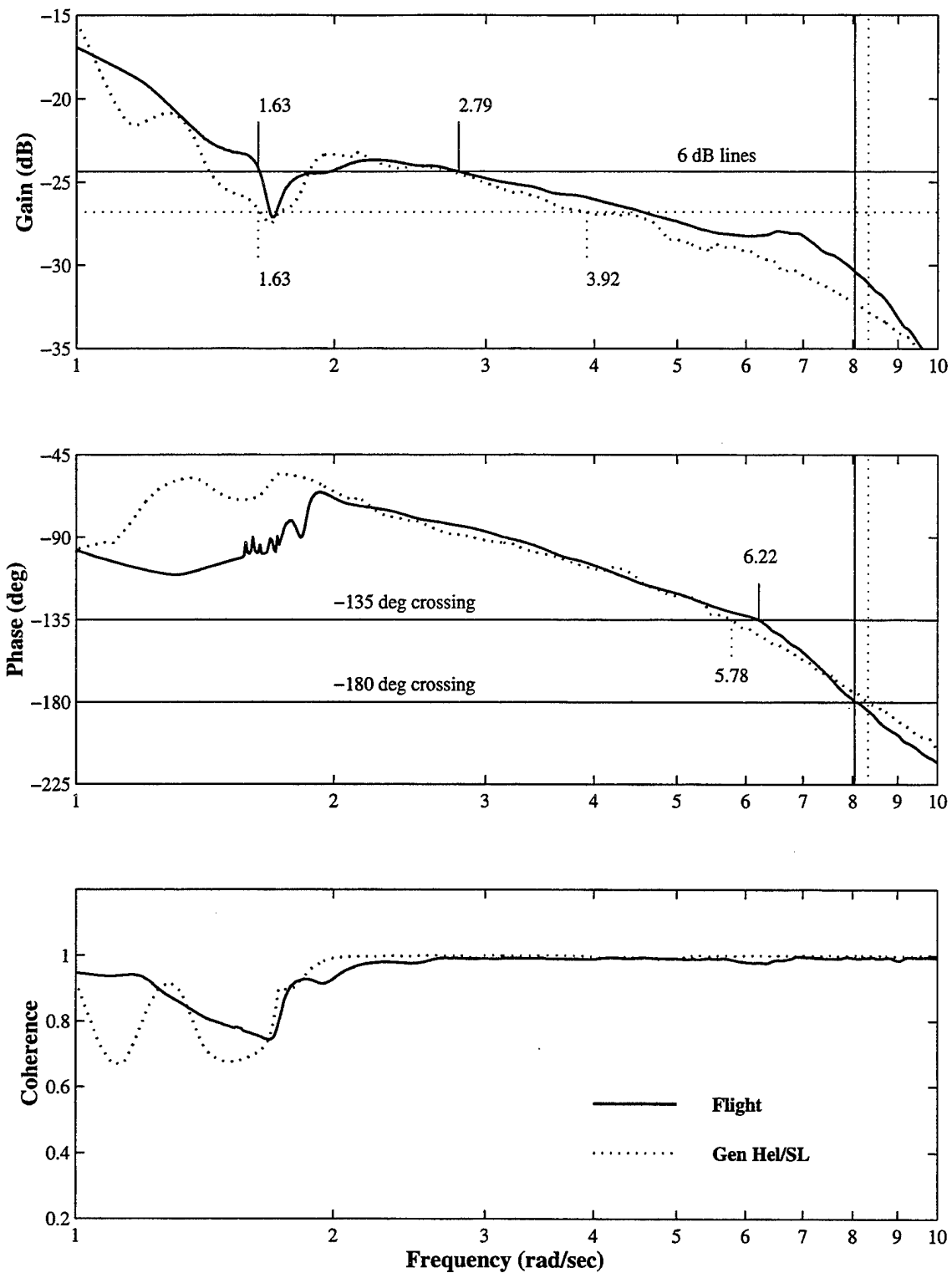


Figure F.2. 4K CONEX Handling Quality Determination, Lateral Axis, (a) Hover



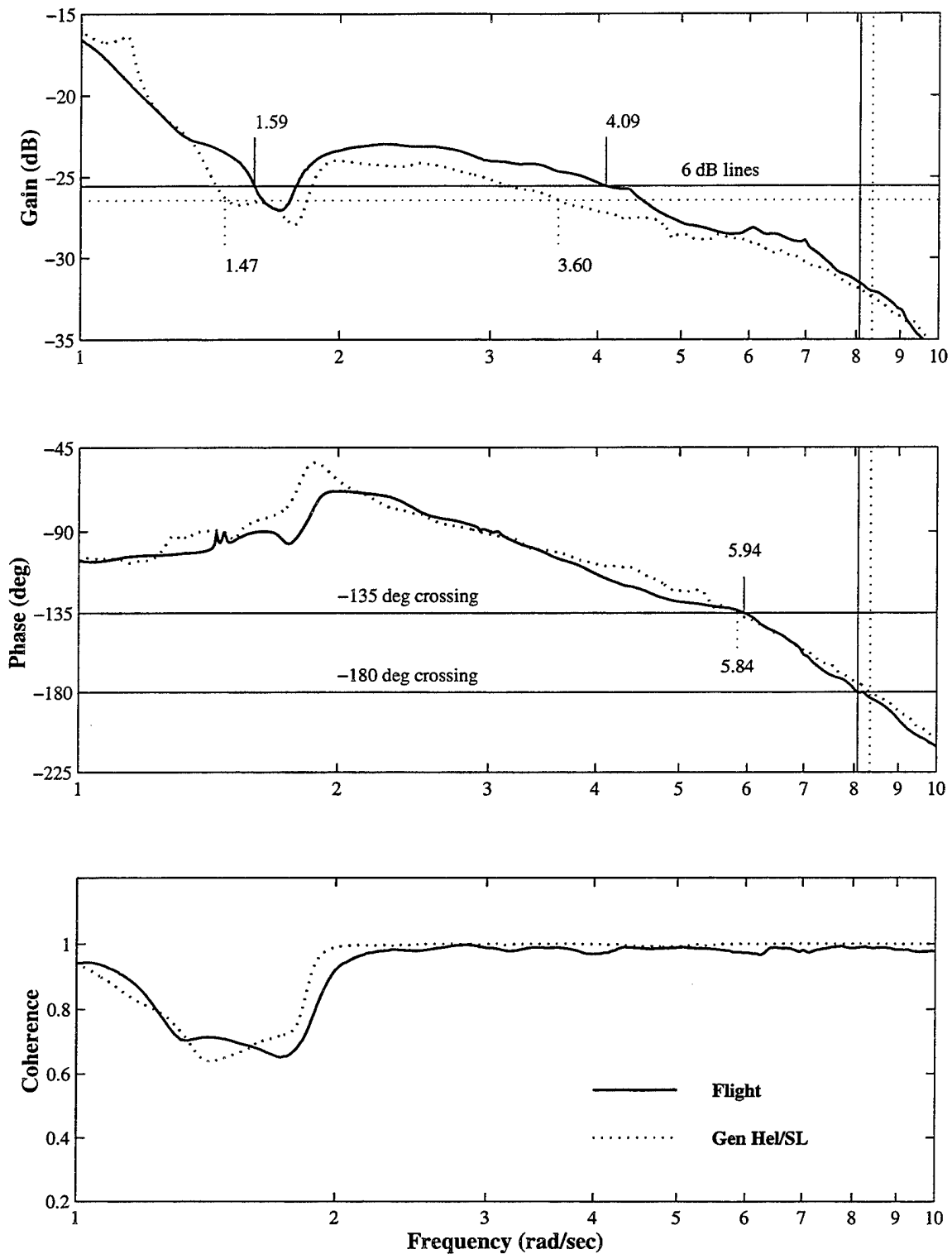


Figure F.2. Continued, (b) 30 Knots

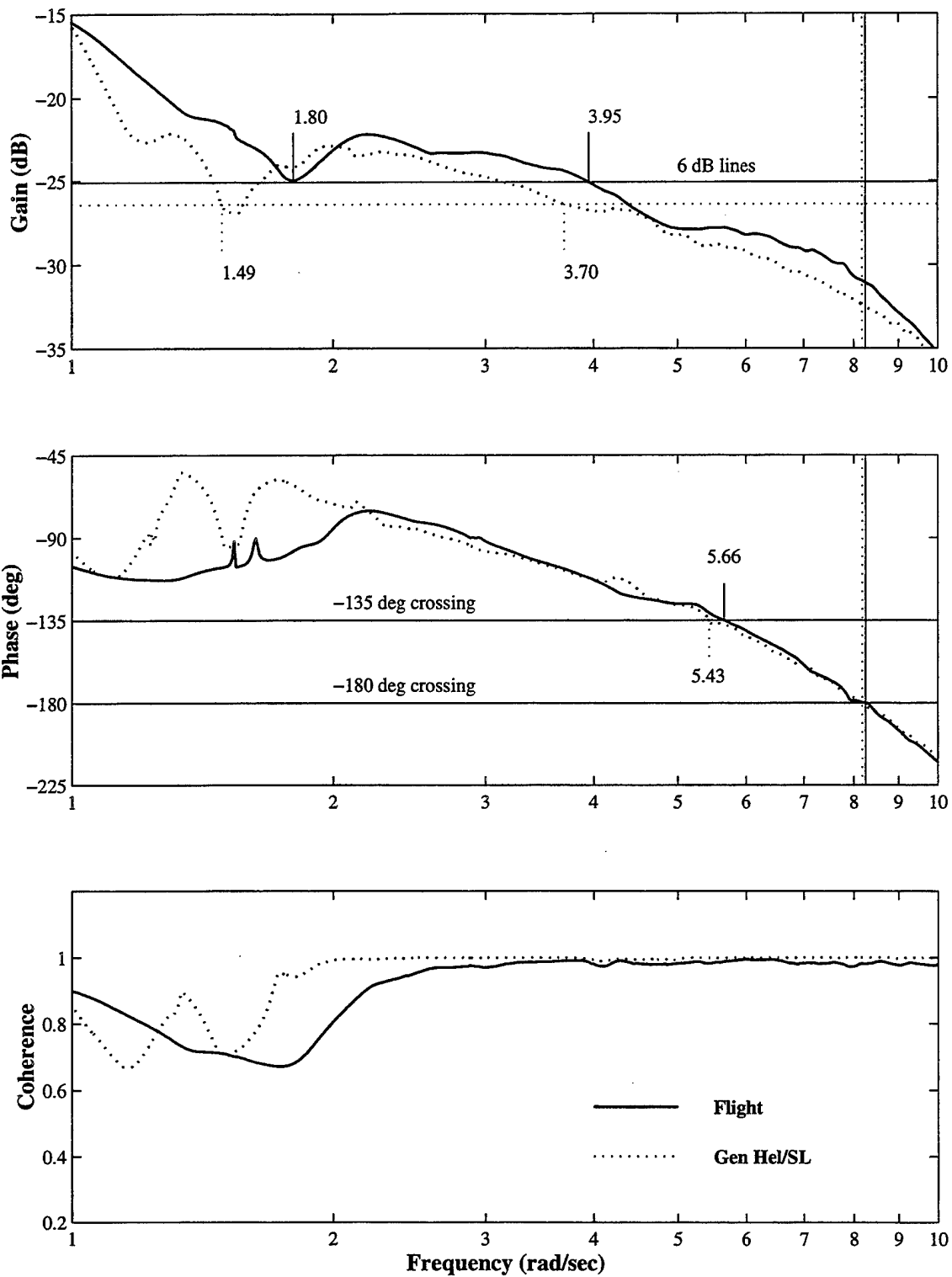


Figure F.2. Continued, (c) 50 Knots

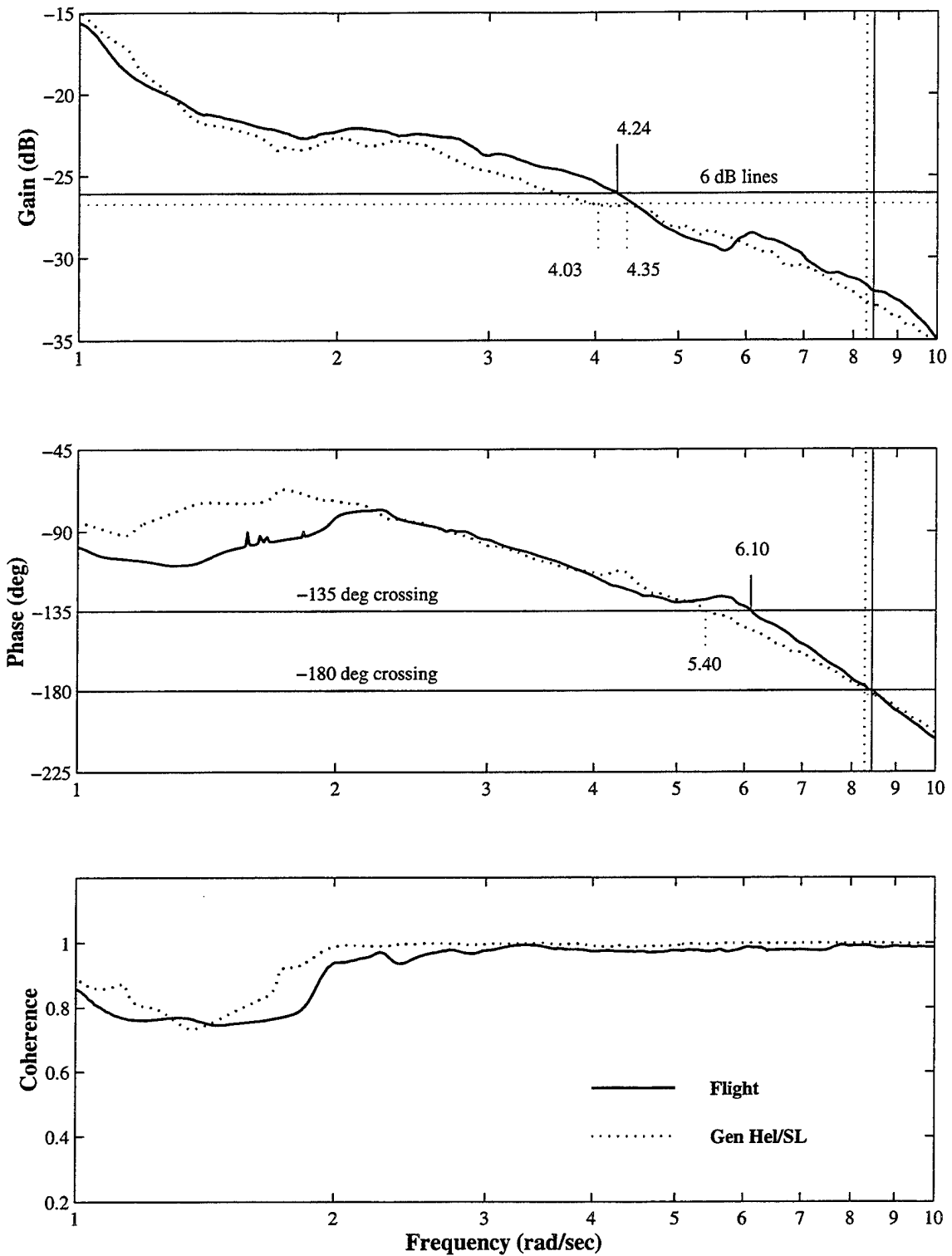


Figure F.2. Continued, (d) 60 Knots

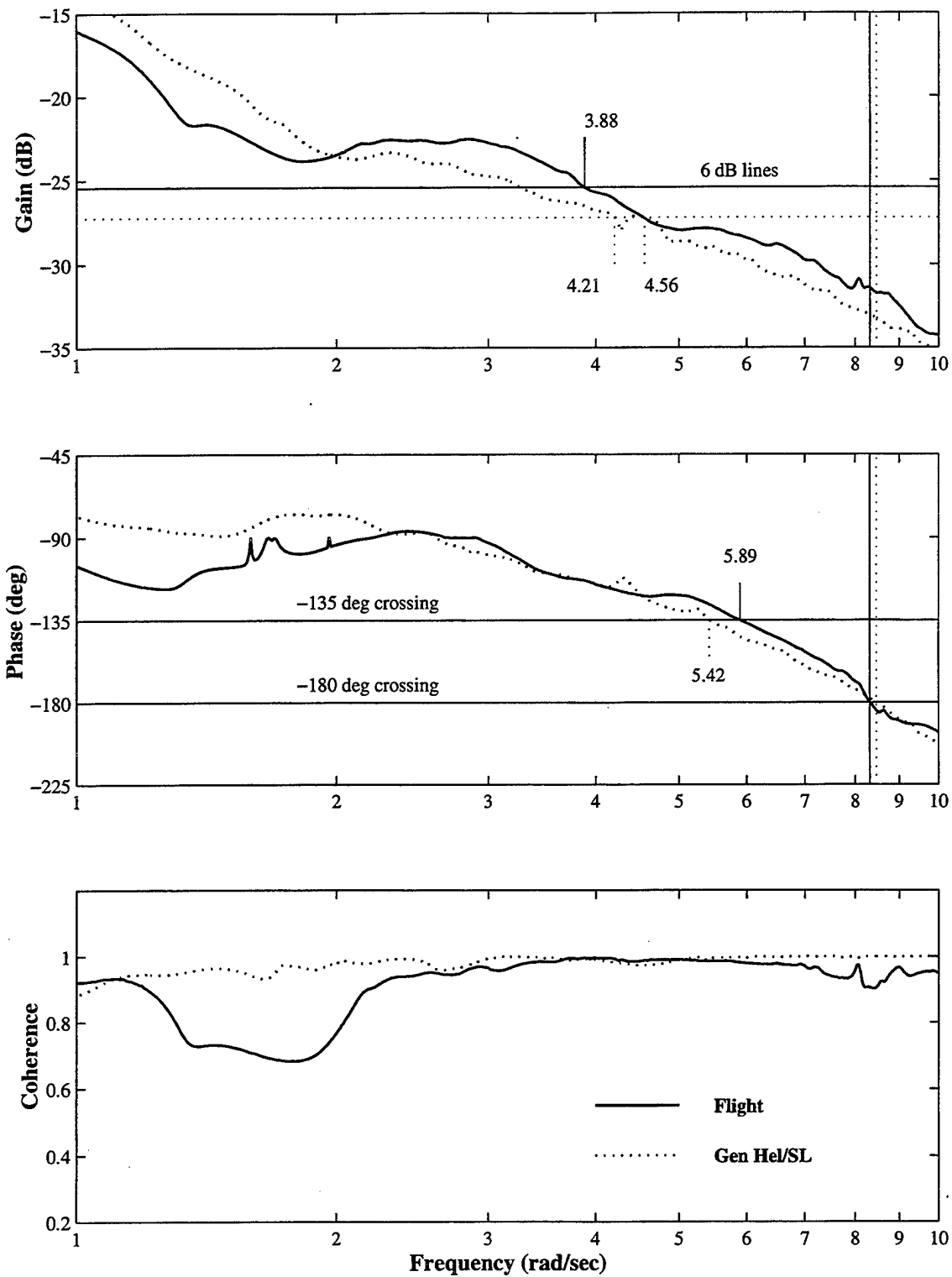


Figure F.2. Continued, (e) 70 Knots

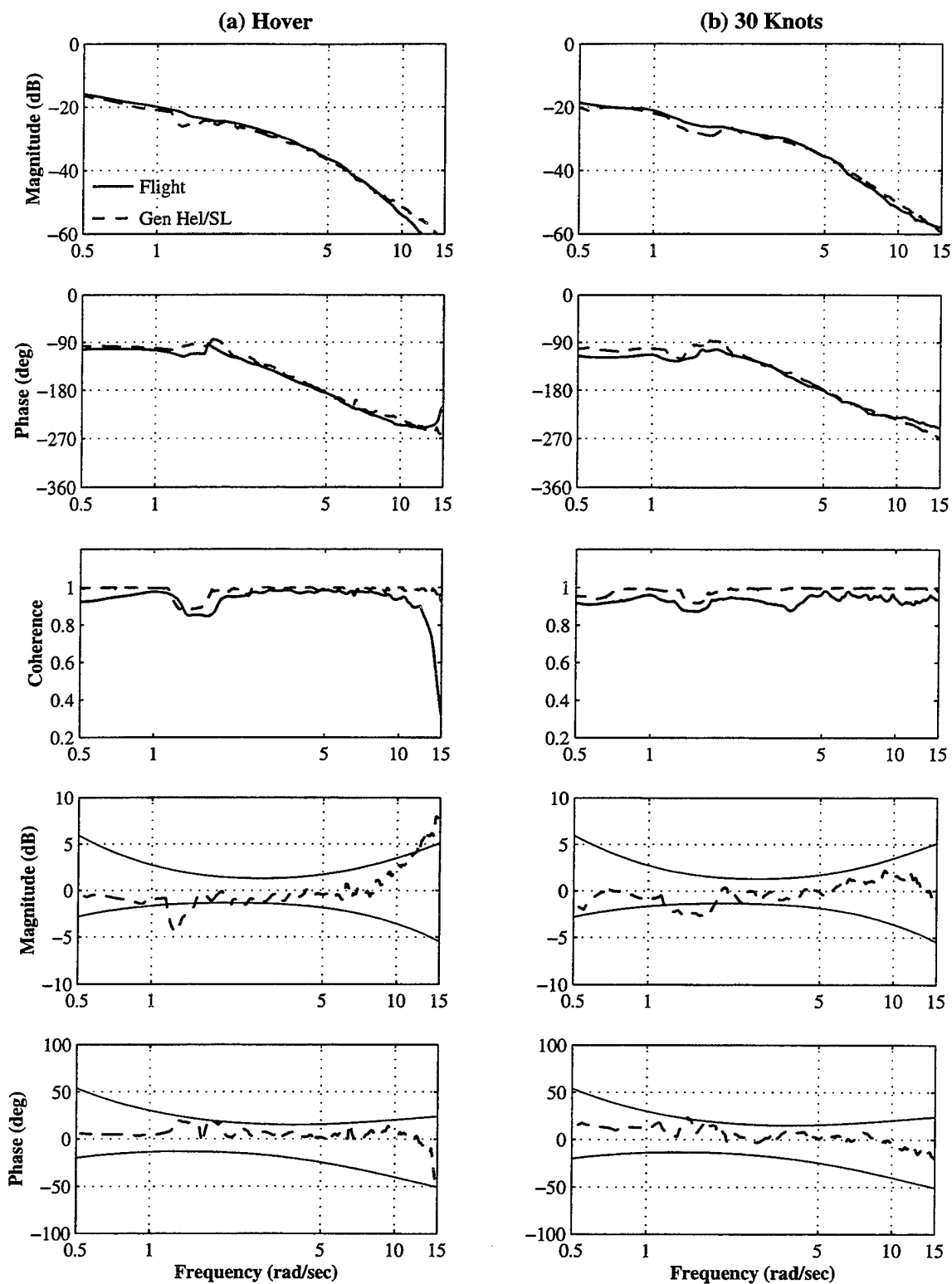


Figure F.3. 4K CONEX Handling Quality, Longitudinal Axis, (a) Hover, (b) 30 Knots

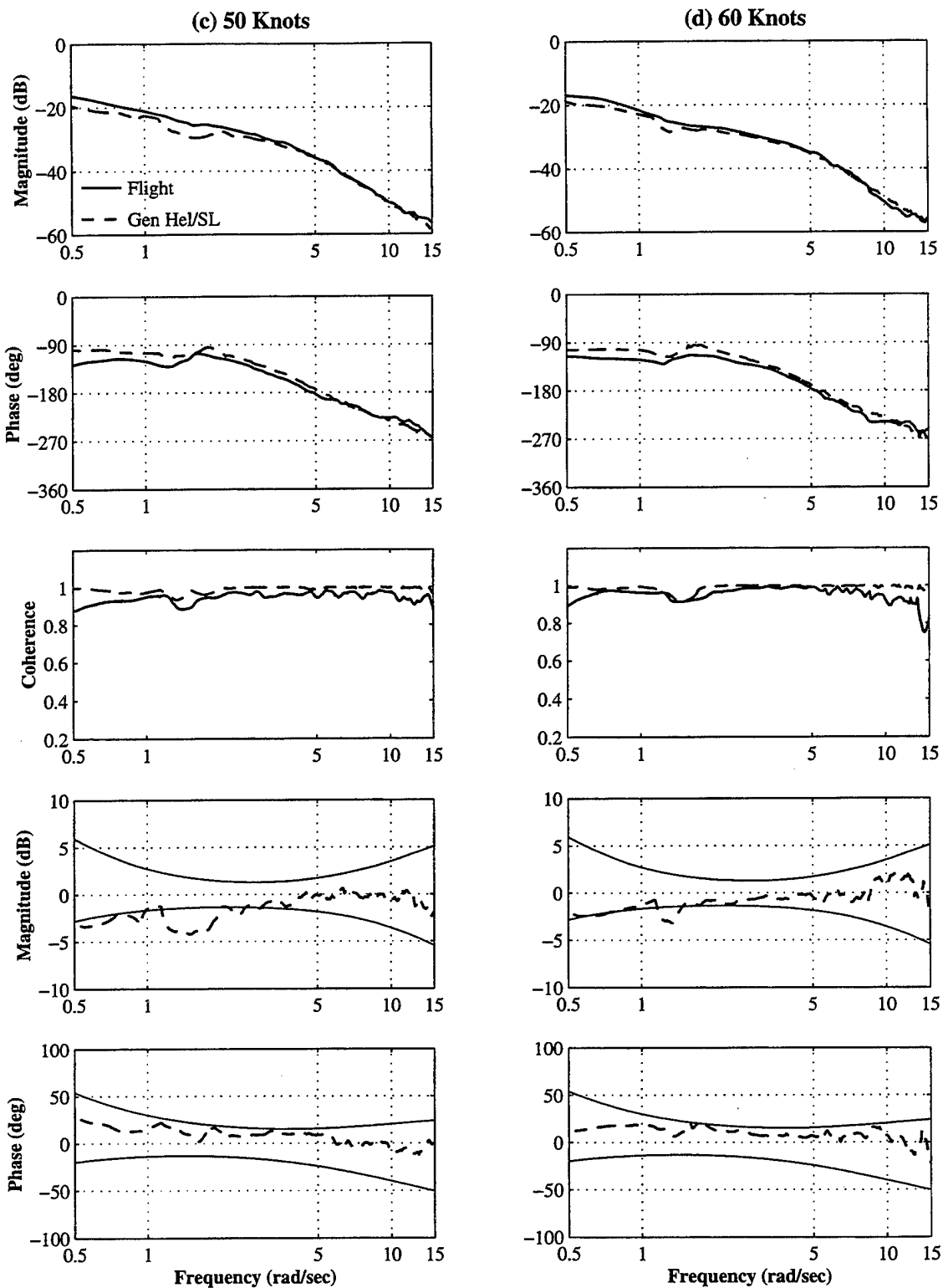


Figure F.3. Continued, (c) 50 Knots, (d) 60 Knots

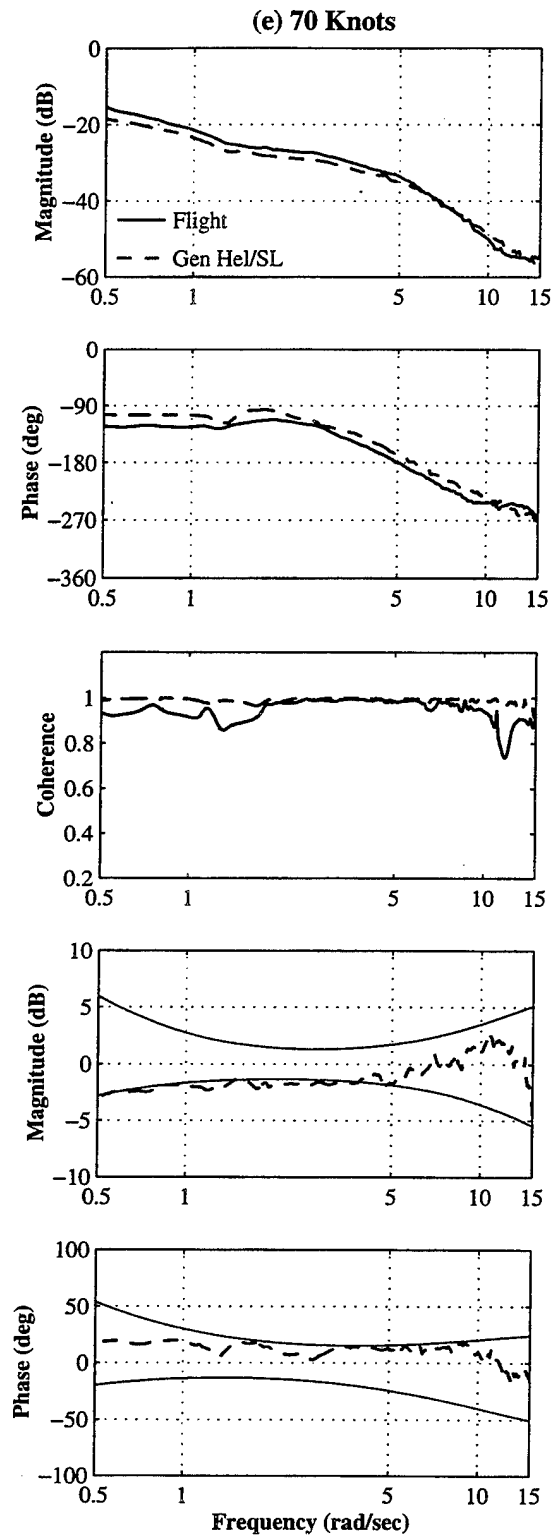


Figure F.3. Continued, (e) 70 Knots

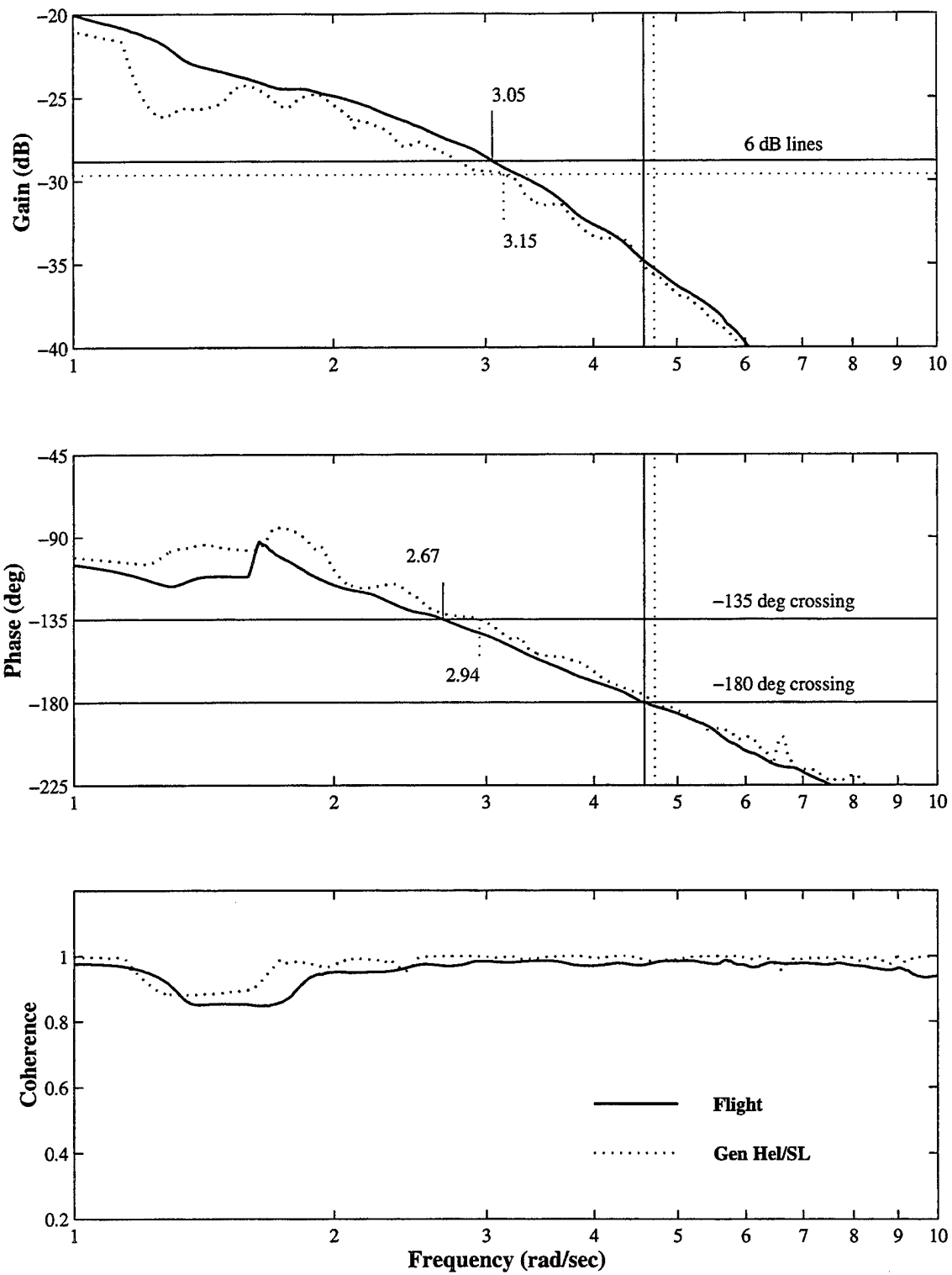


Figure F.4. 4K CONEX Handling Quality Determination, Longitudinal Axis, (a) Hover



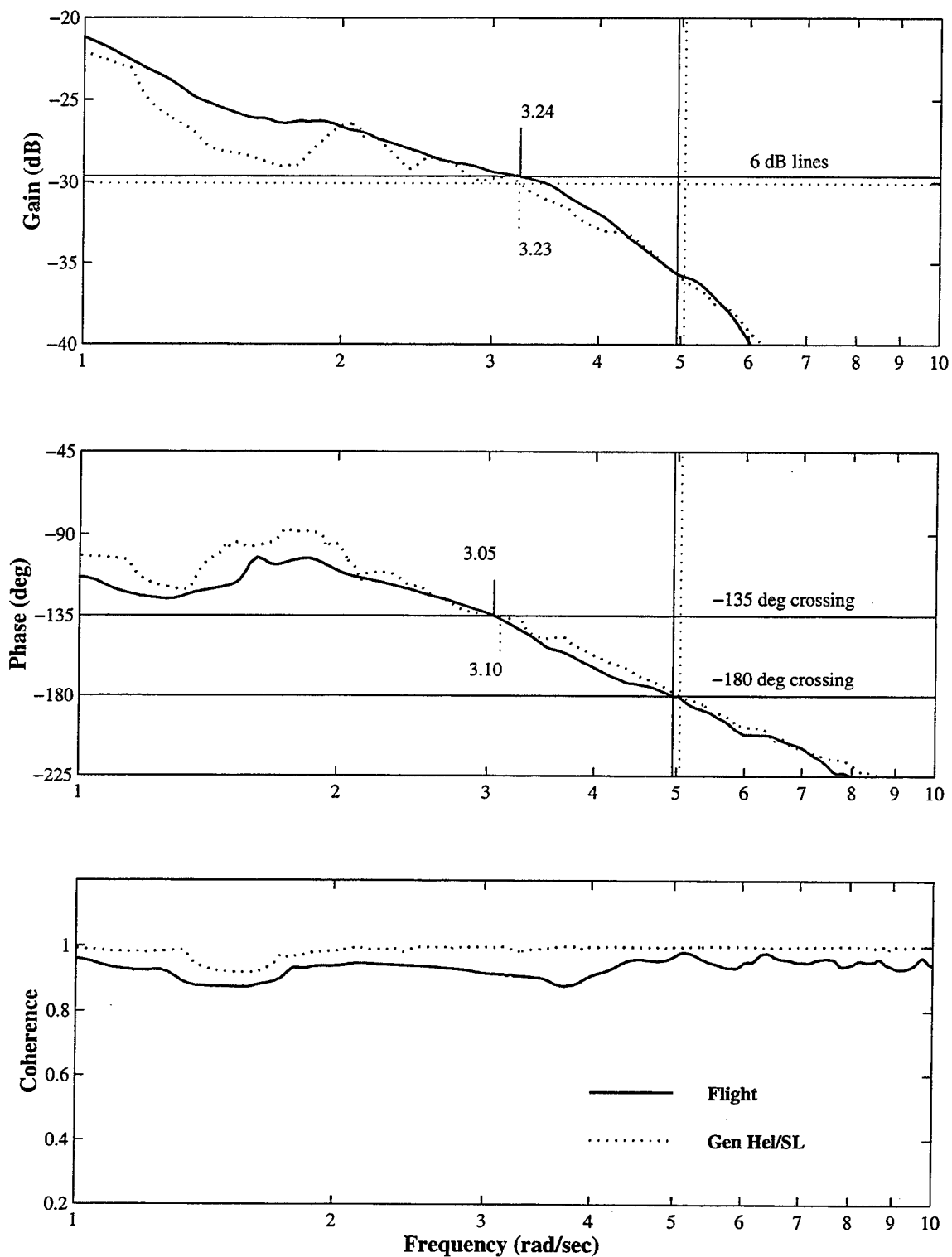


Figure F.4. Continued, (b) 30 Knots

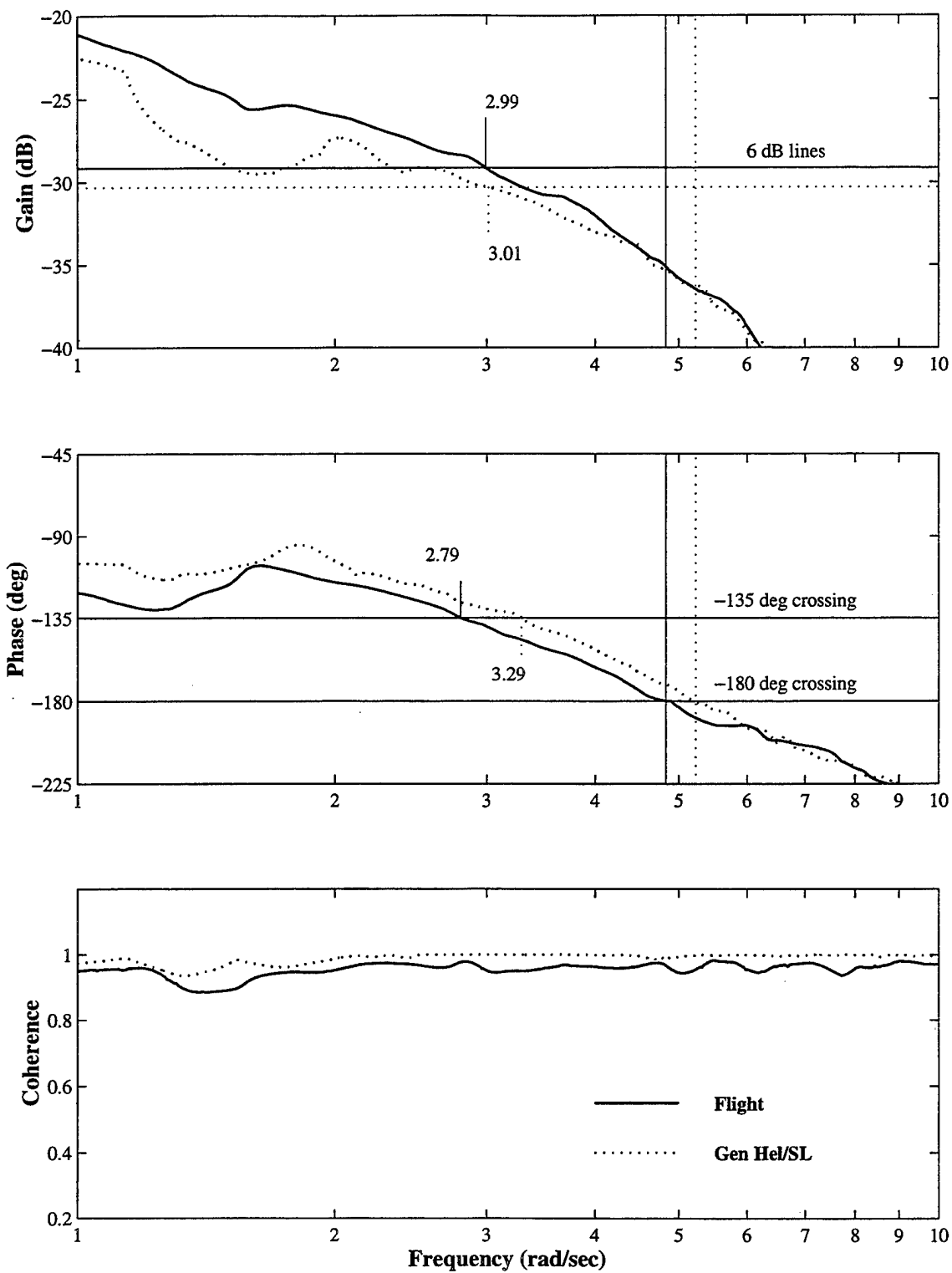


Figure F.4. Continued, (c) 50 Knots

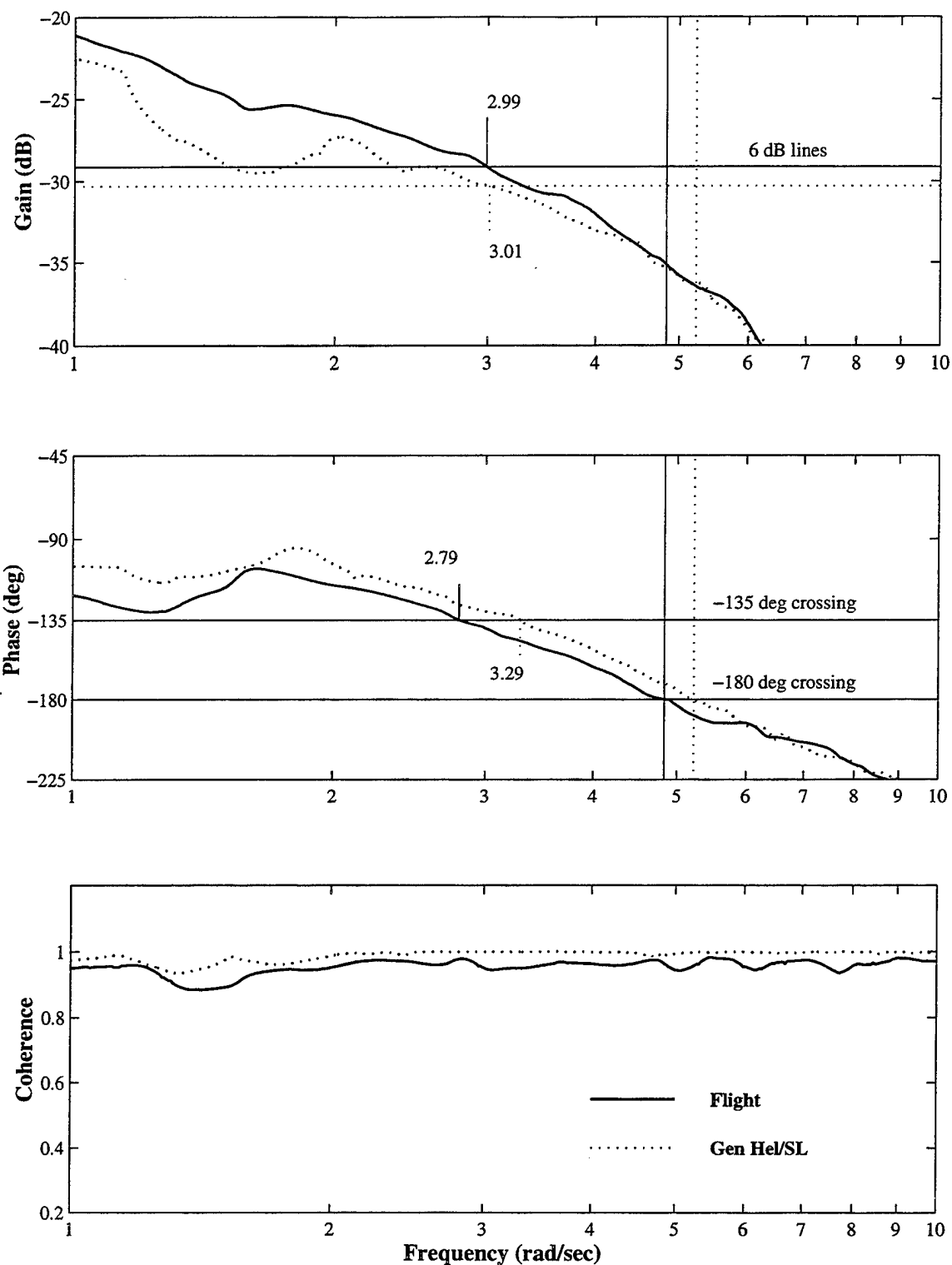


Figure F.4. Continued, (d) 60 Knots

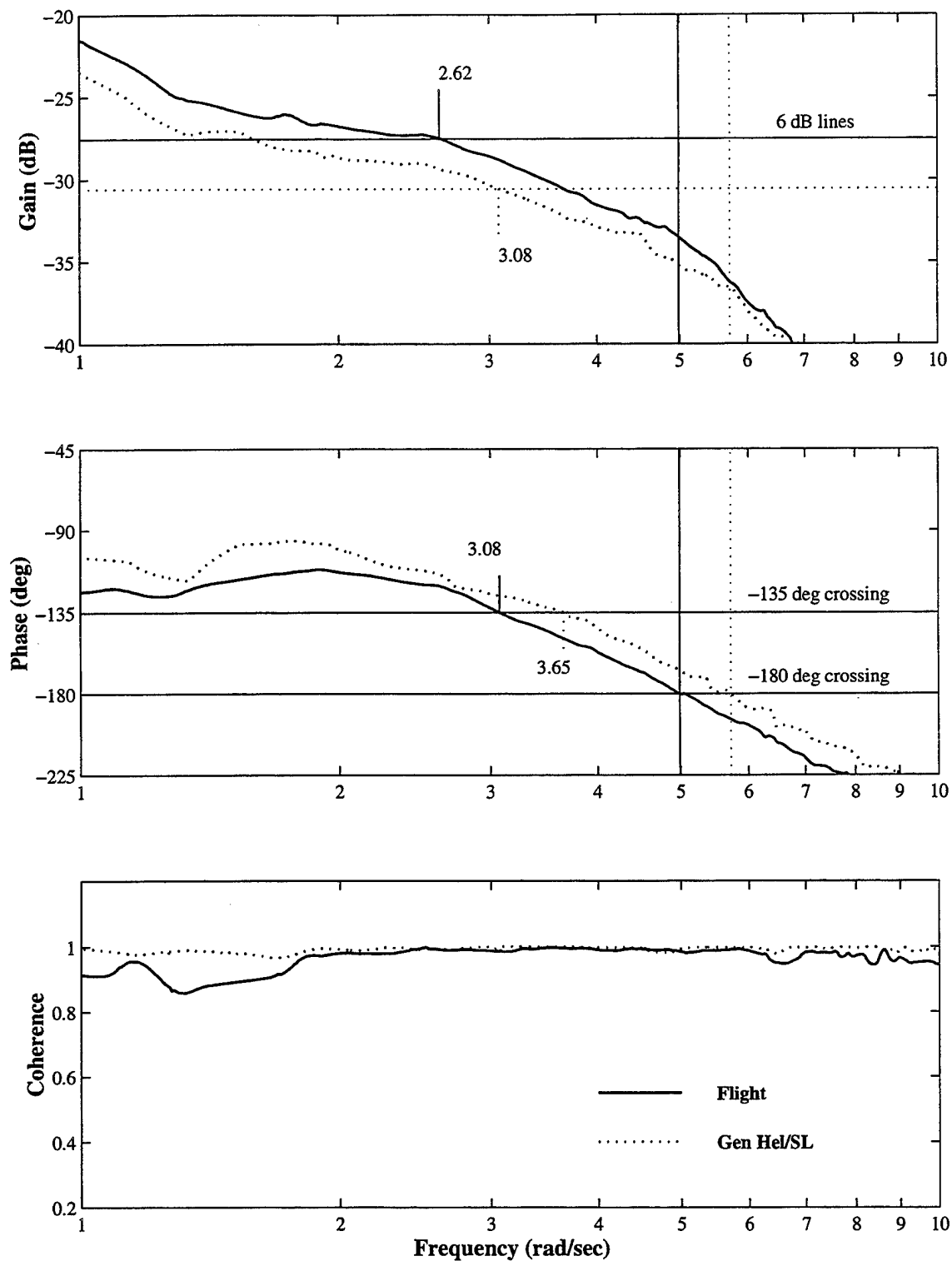


Figure F.4. Continued, (e) 70 Knots

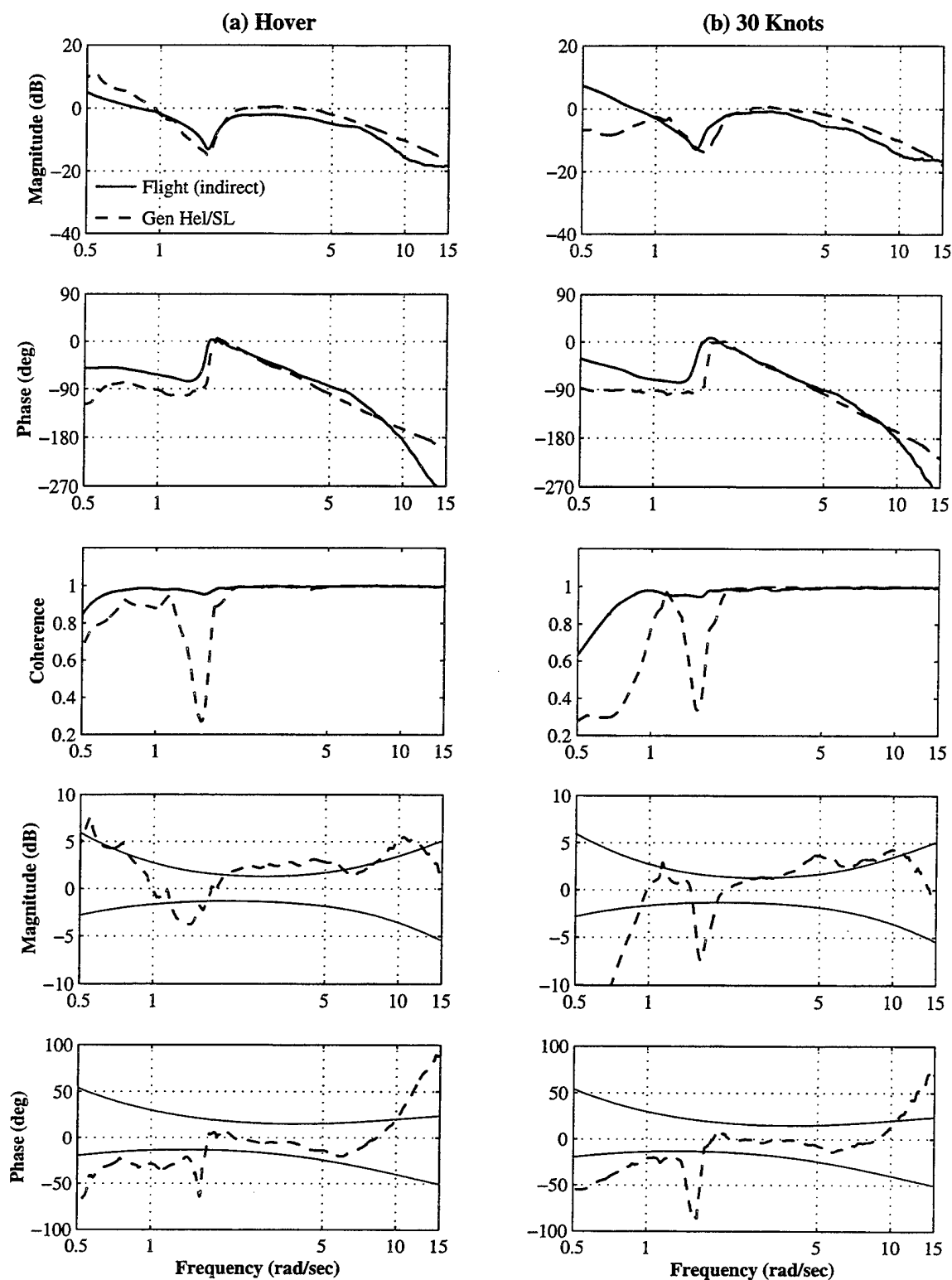


Figure F.5. 4K CONEX Stability Margin, Lateral Axis, (a) Hover, (b) 30 Knots

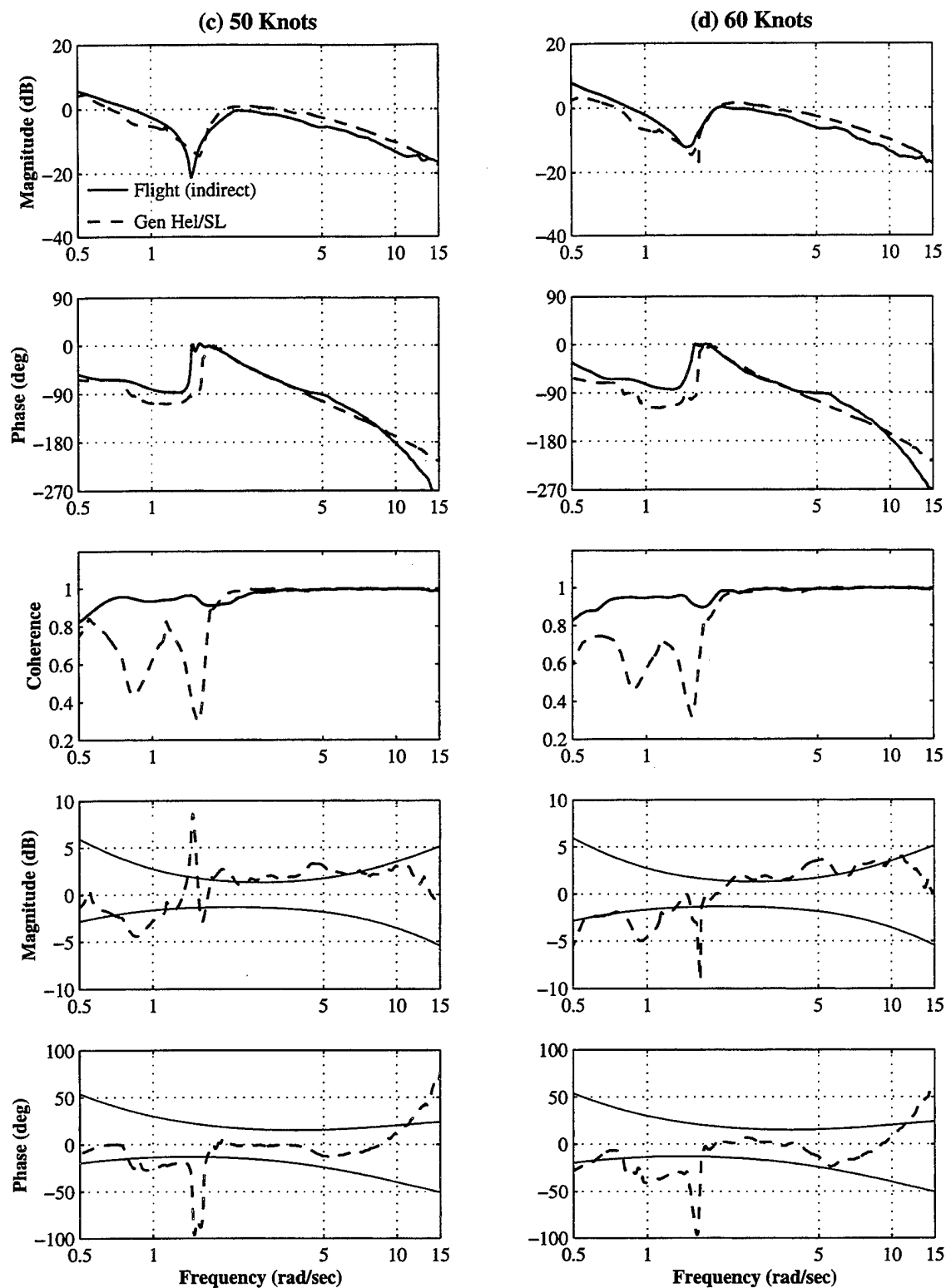


Figure F.5. Continued, (c) 50 Knots, (d) 60 Knots

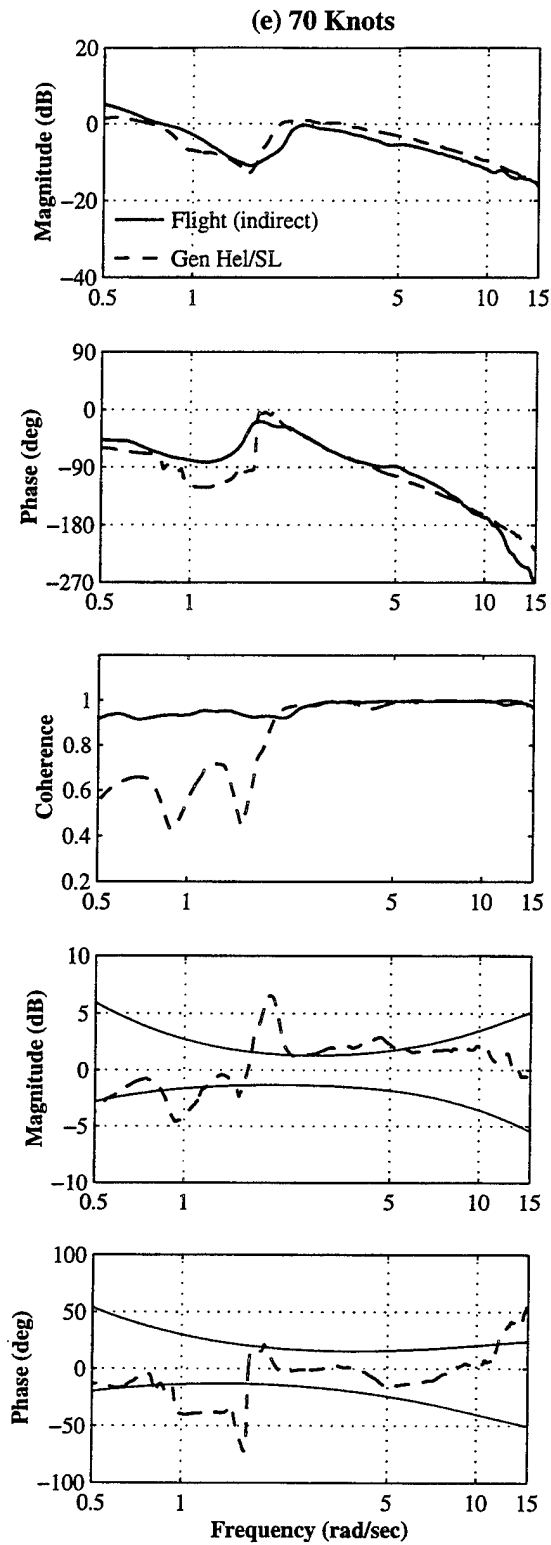


Figure F.5. Continued, (e) 70 Knots

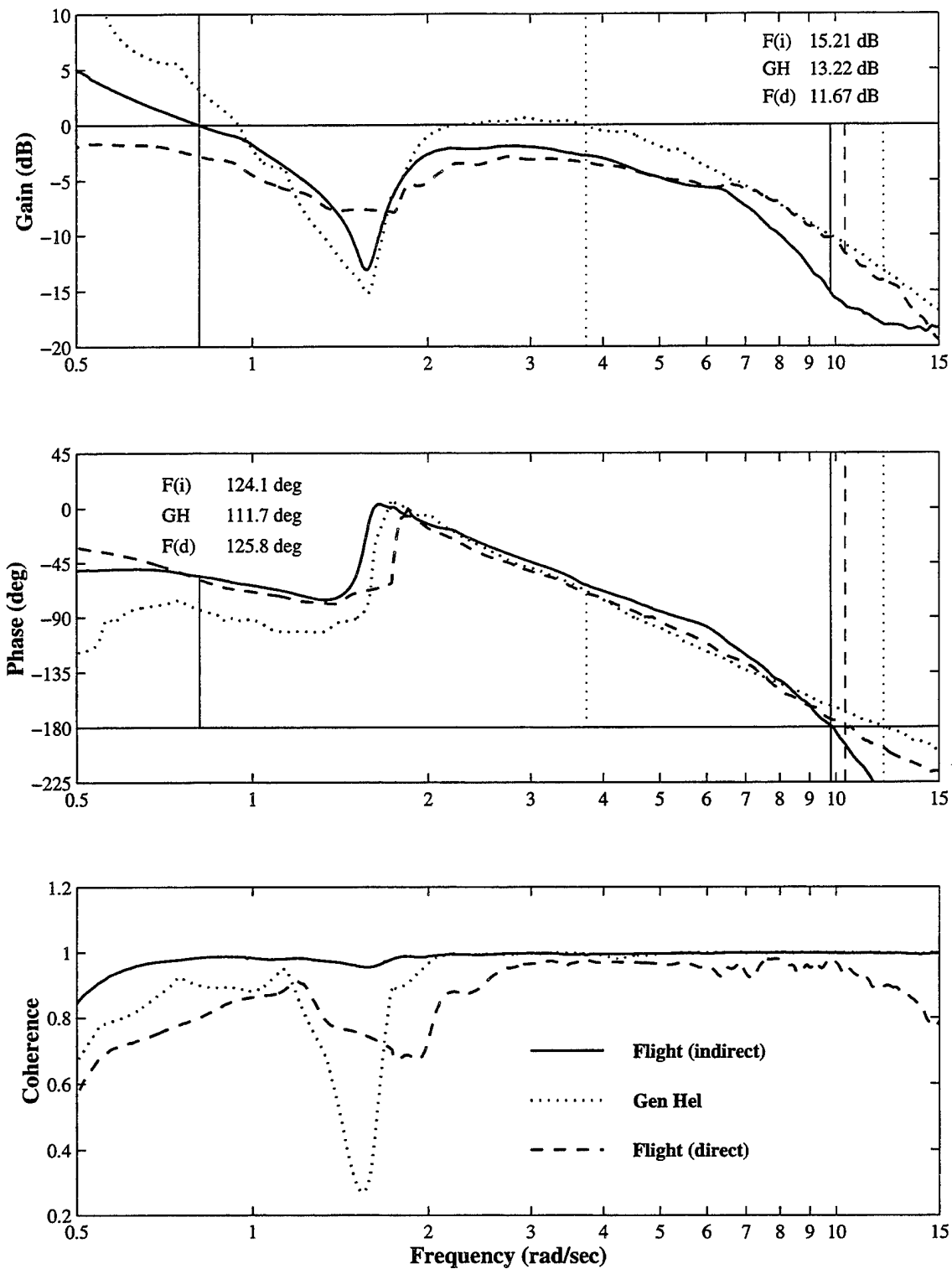


Figure F.6. 4K CONEX Stability Margin Determination, Lateral Axis, (a) Hover



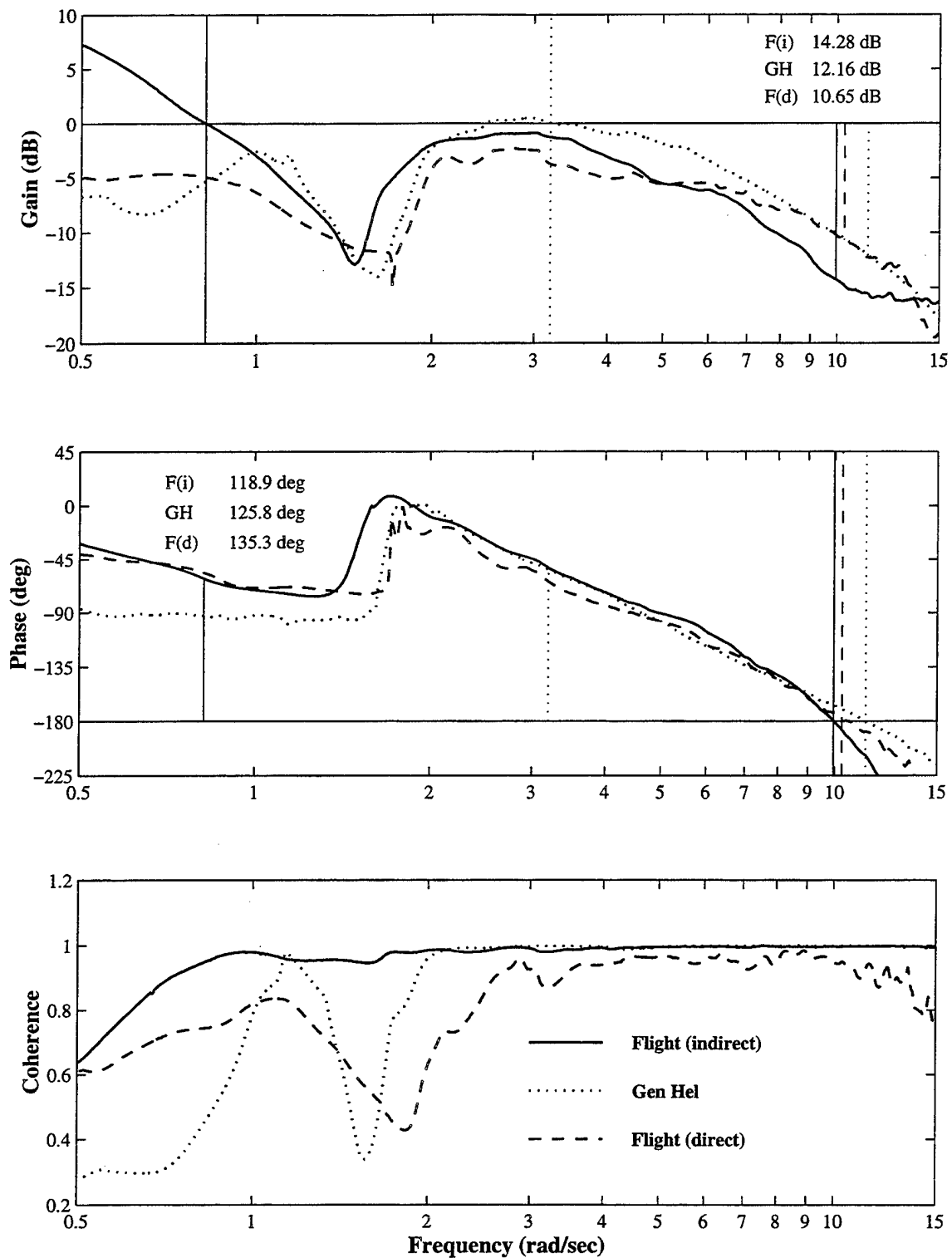


Figure F.6. Continued, (b) 30 Knots

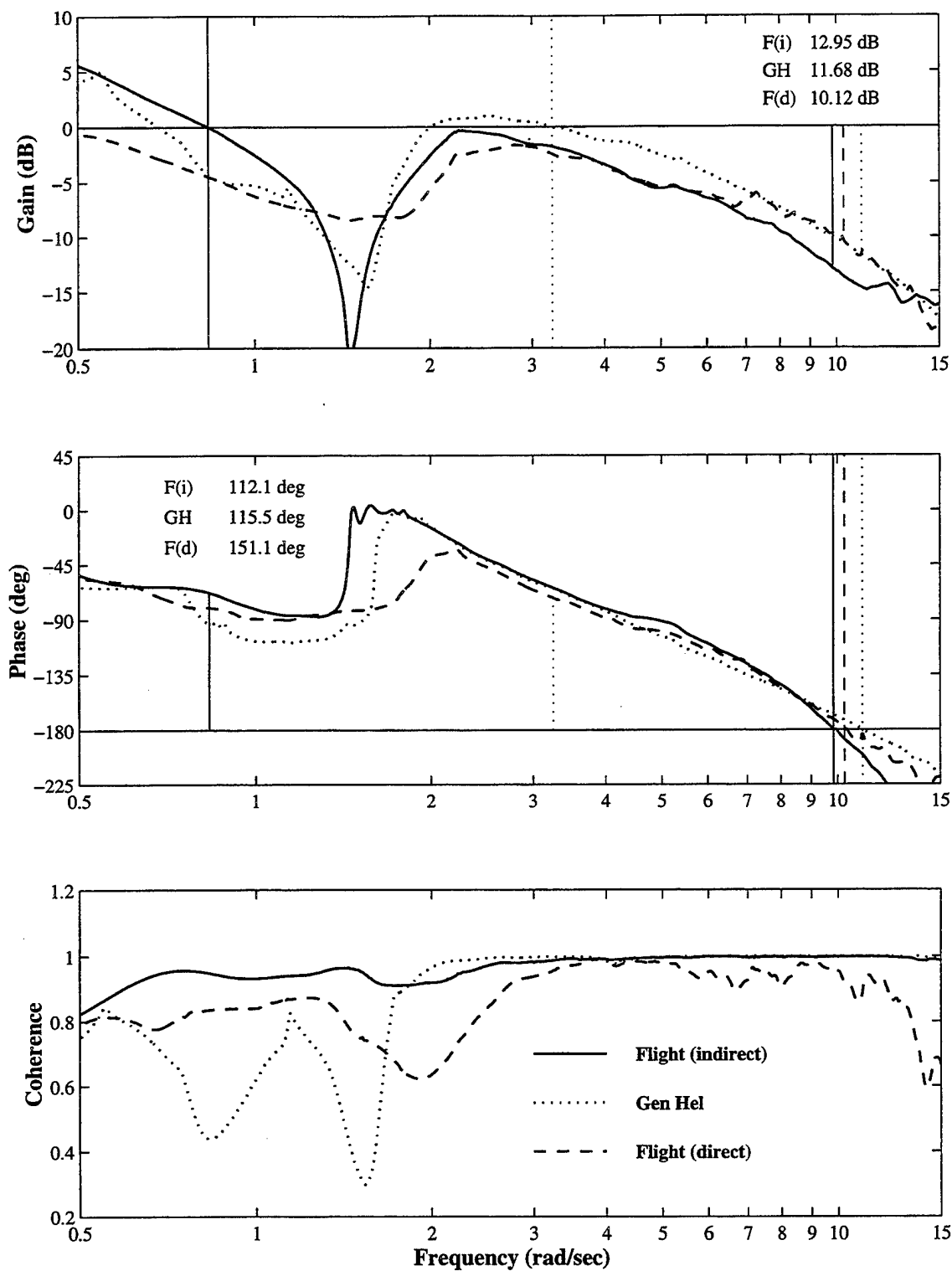


Figure F.6. Continued, (c) 50 Knots

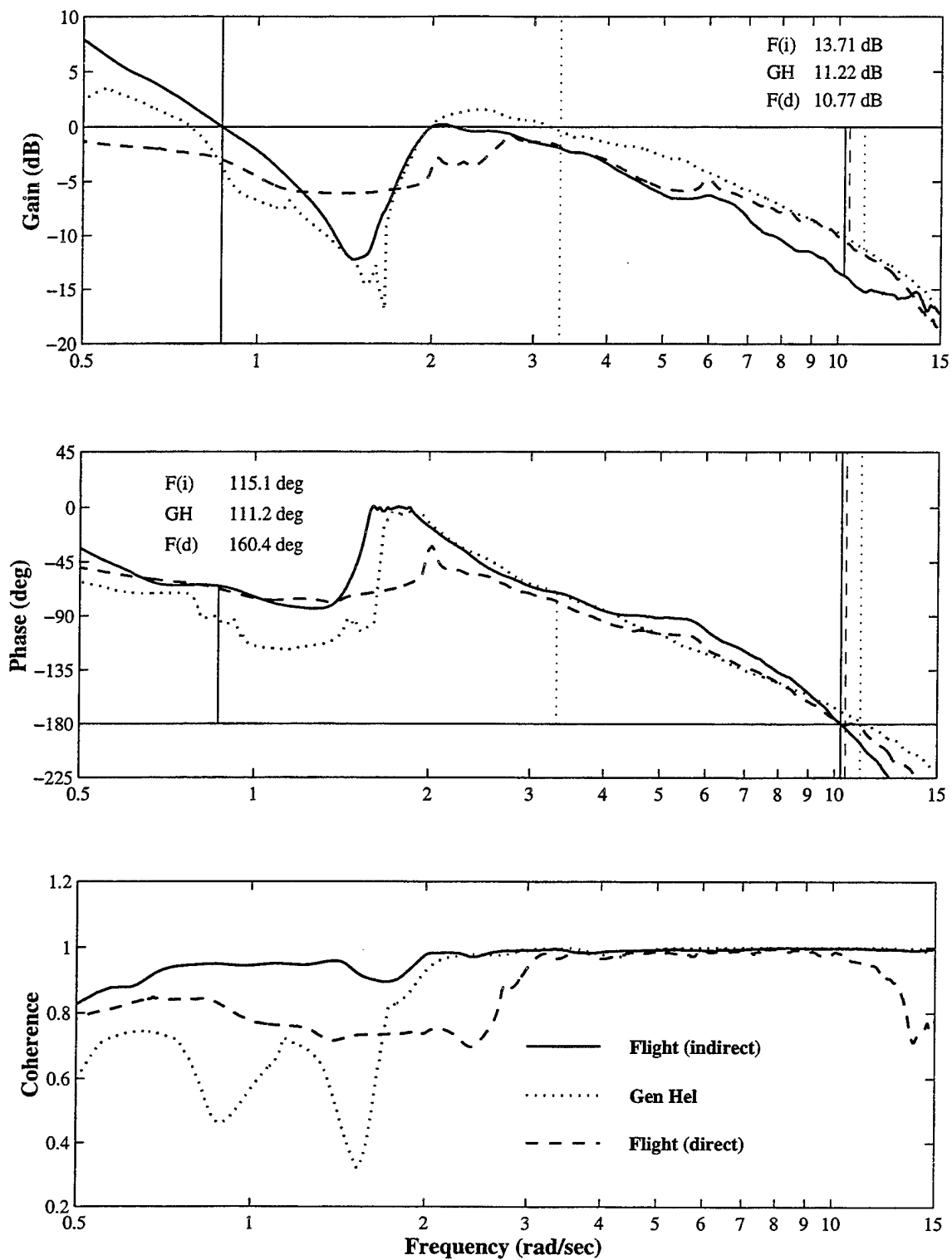


Figure F.6. Continued, (d) 60 Knots

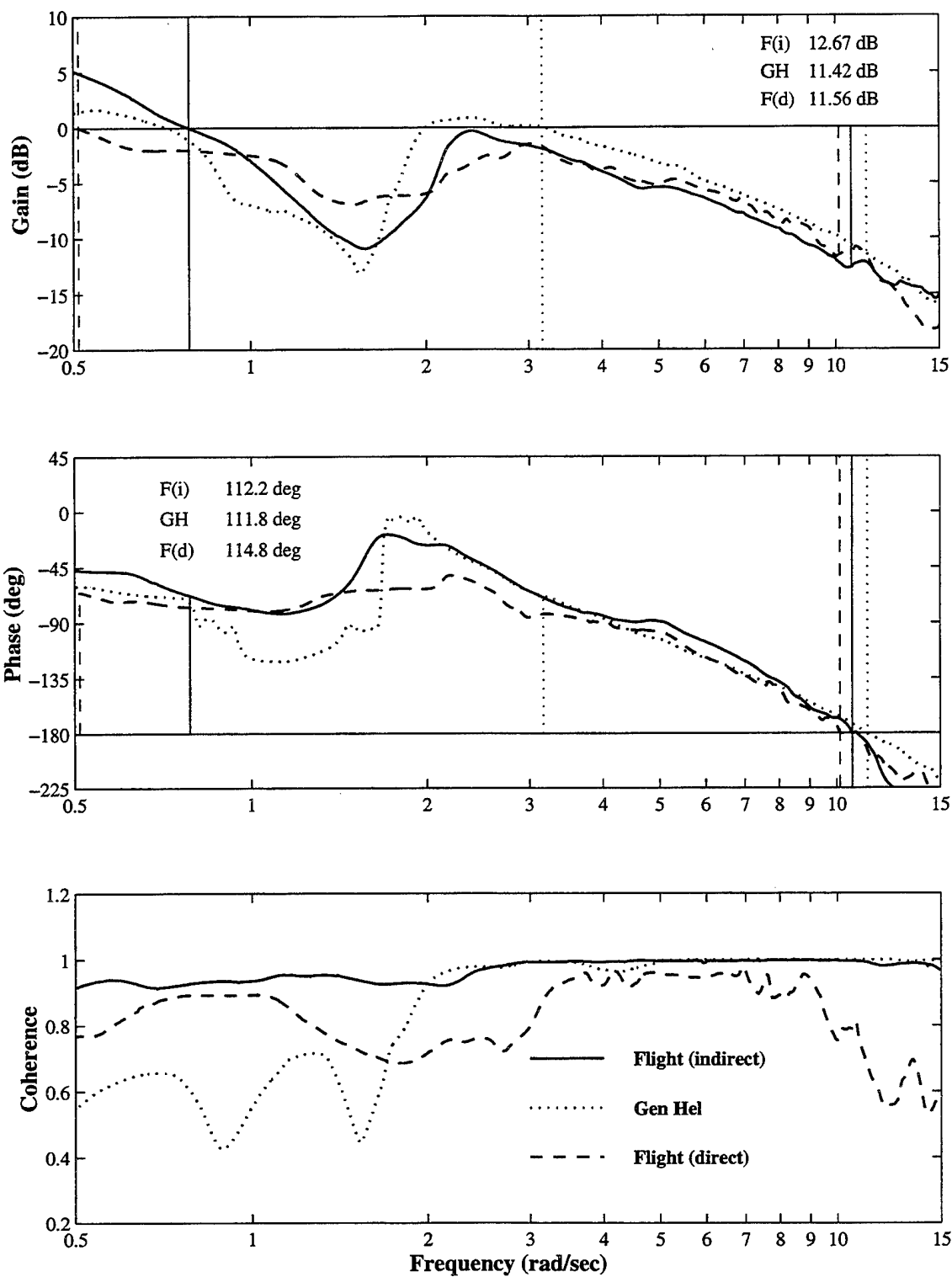


Figure F.6. Continued, (e) 70 Knots

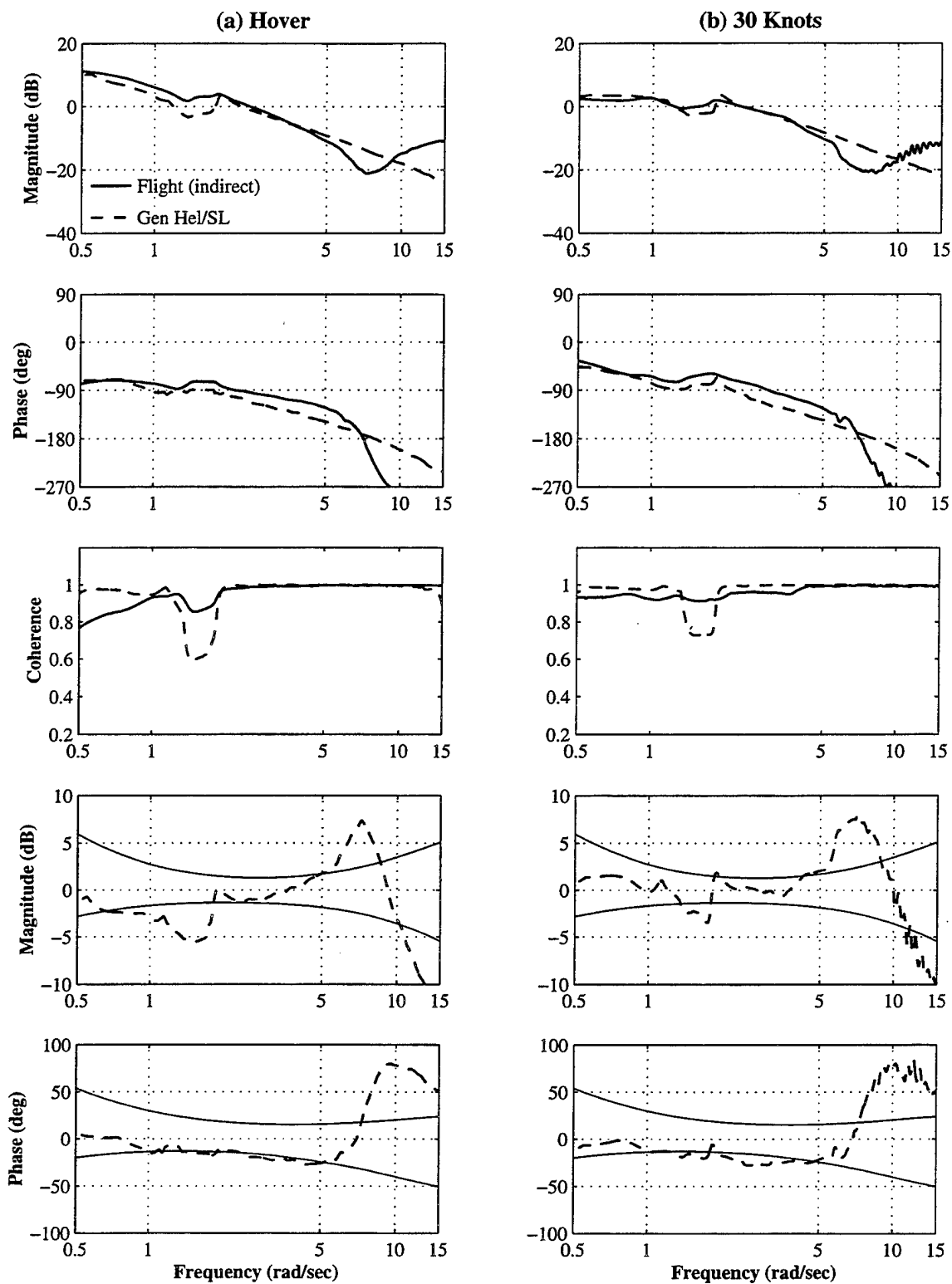


Figure F.7. 4K CONEX Stability Margin, Longitudinal Axis, (a) Hover, (b) 30 Knots

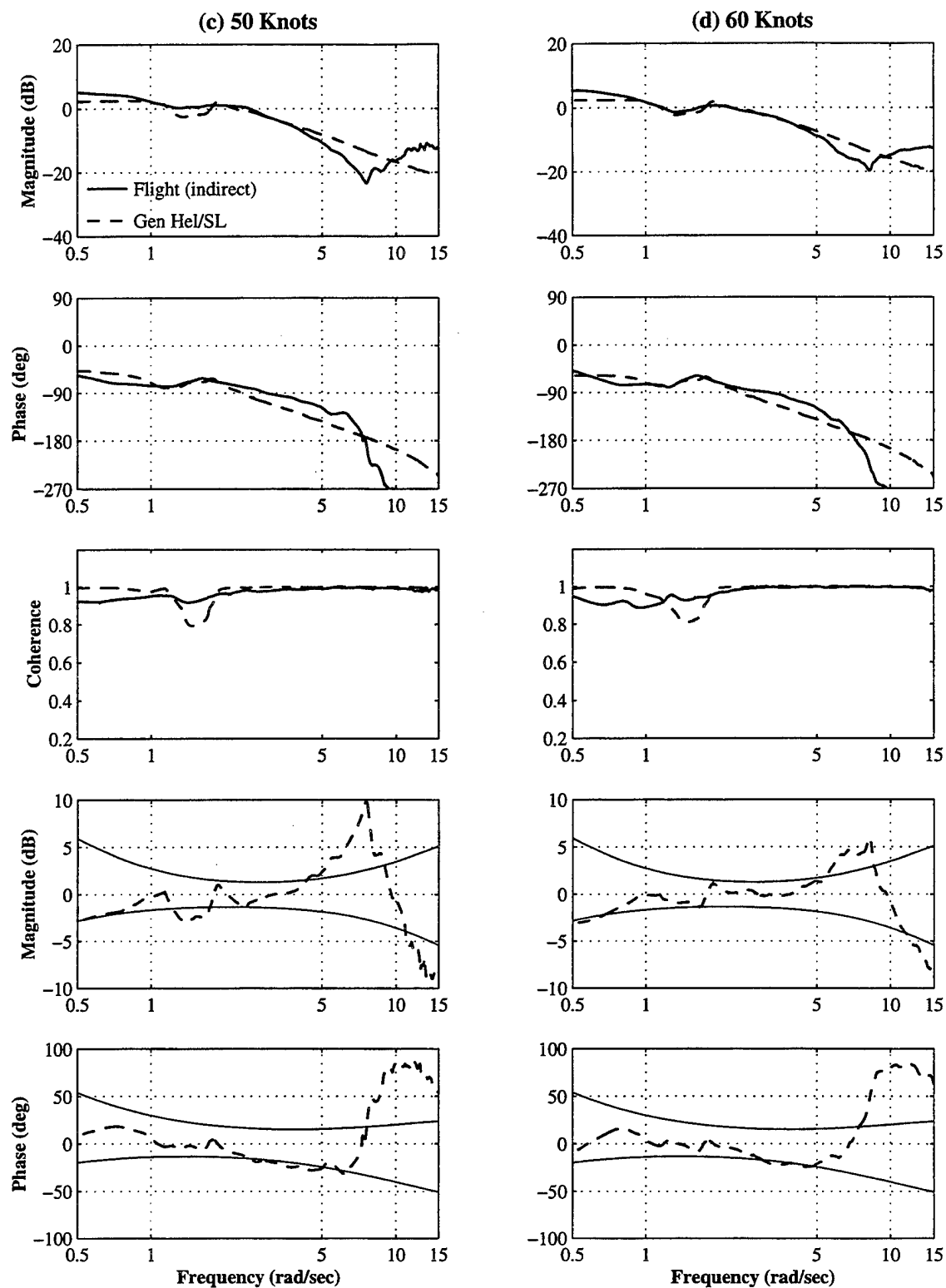


Figure F.7. Continued, (c) 50 Knots, (d) 60 Knots

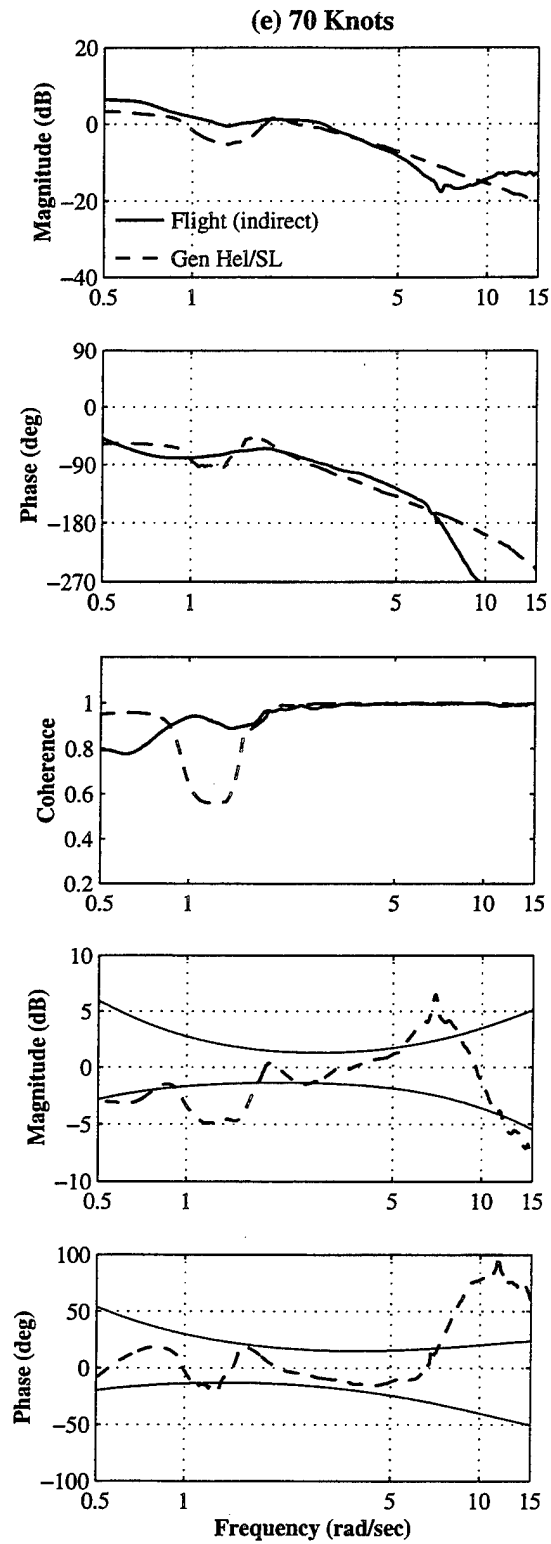


Figure F.7. Continued, (e) 70 Knots

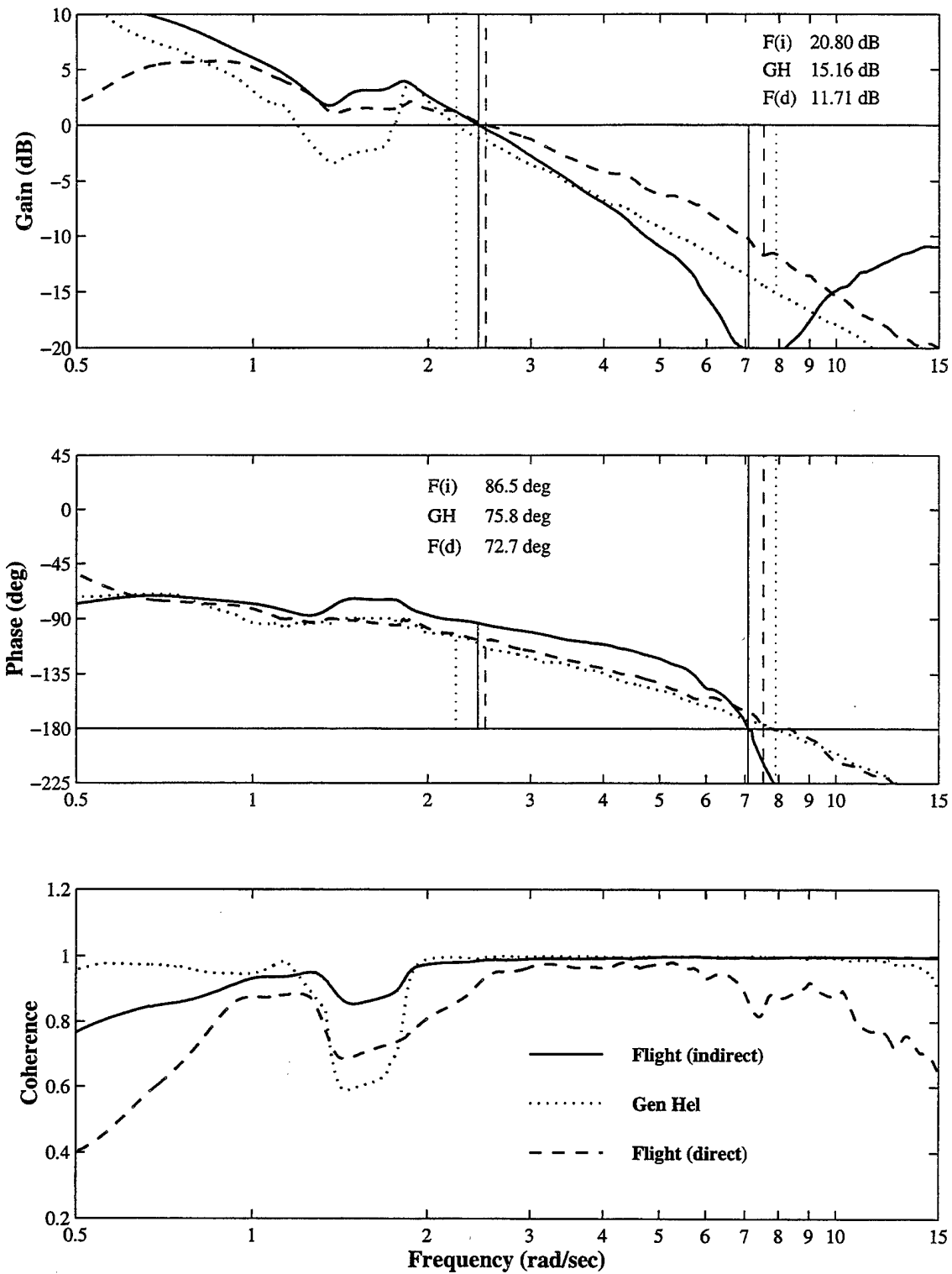


Figure F.8. 4K CONEX Stability Margin Determination, Longitudinal Axis, (a) Hover



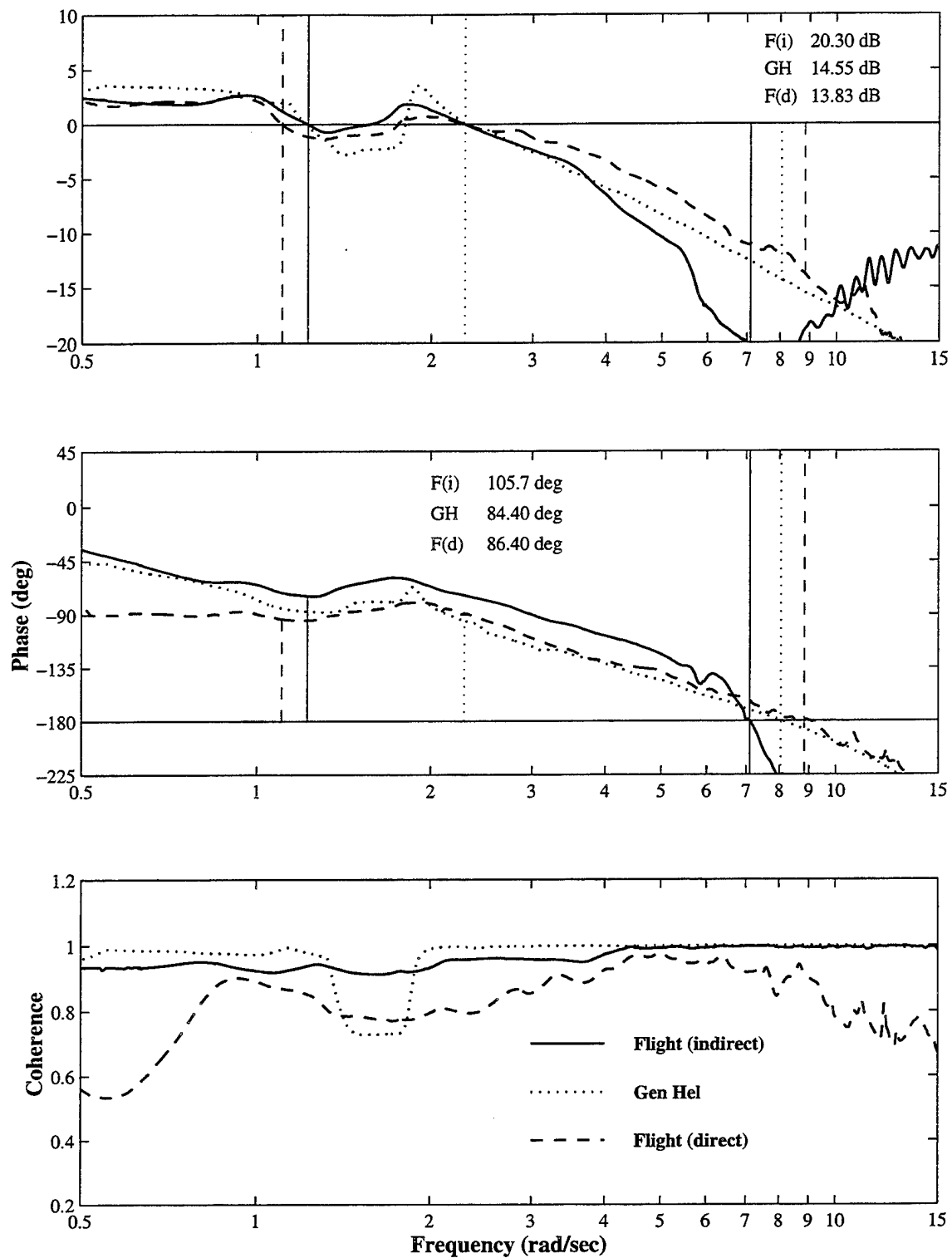


Figure F.8. Continued, (b) 30 Knots

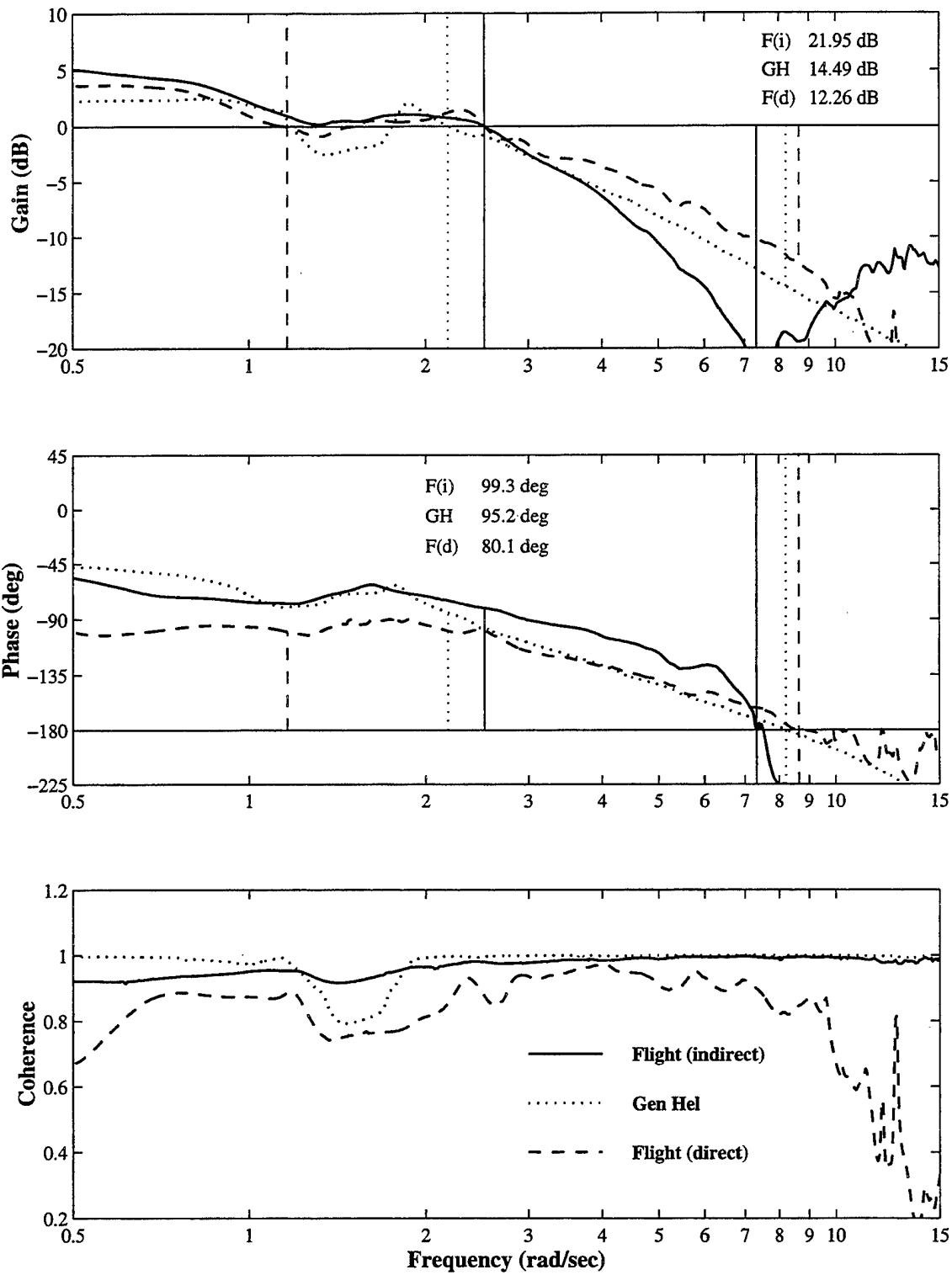


Figure F.8. Continued, (c) 50 Knots

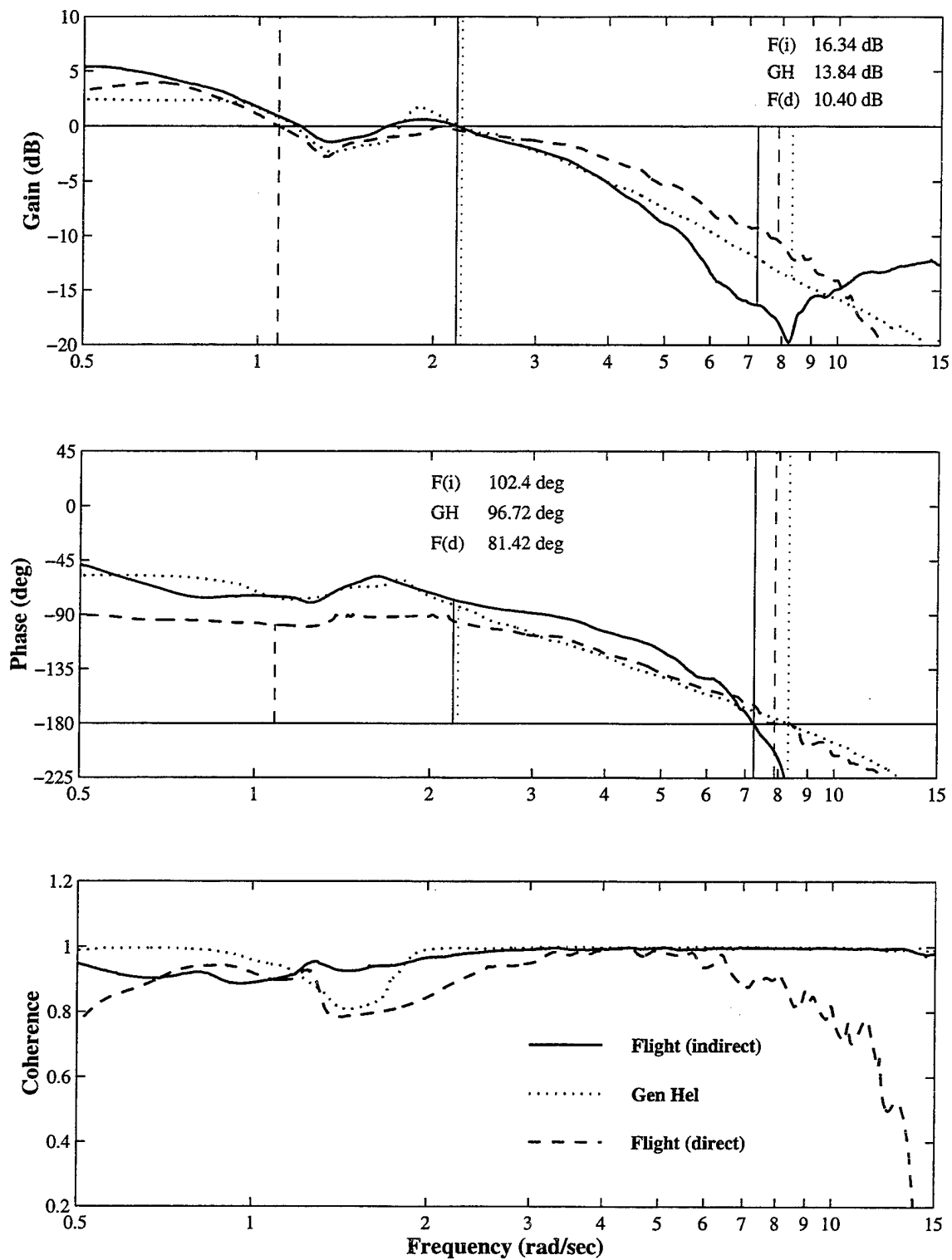


Figure F.8. Continued, (d) 60 Knots

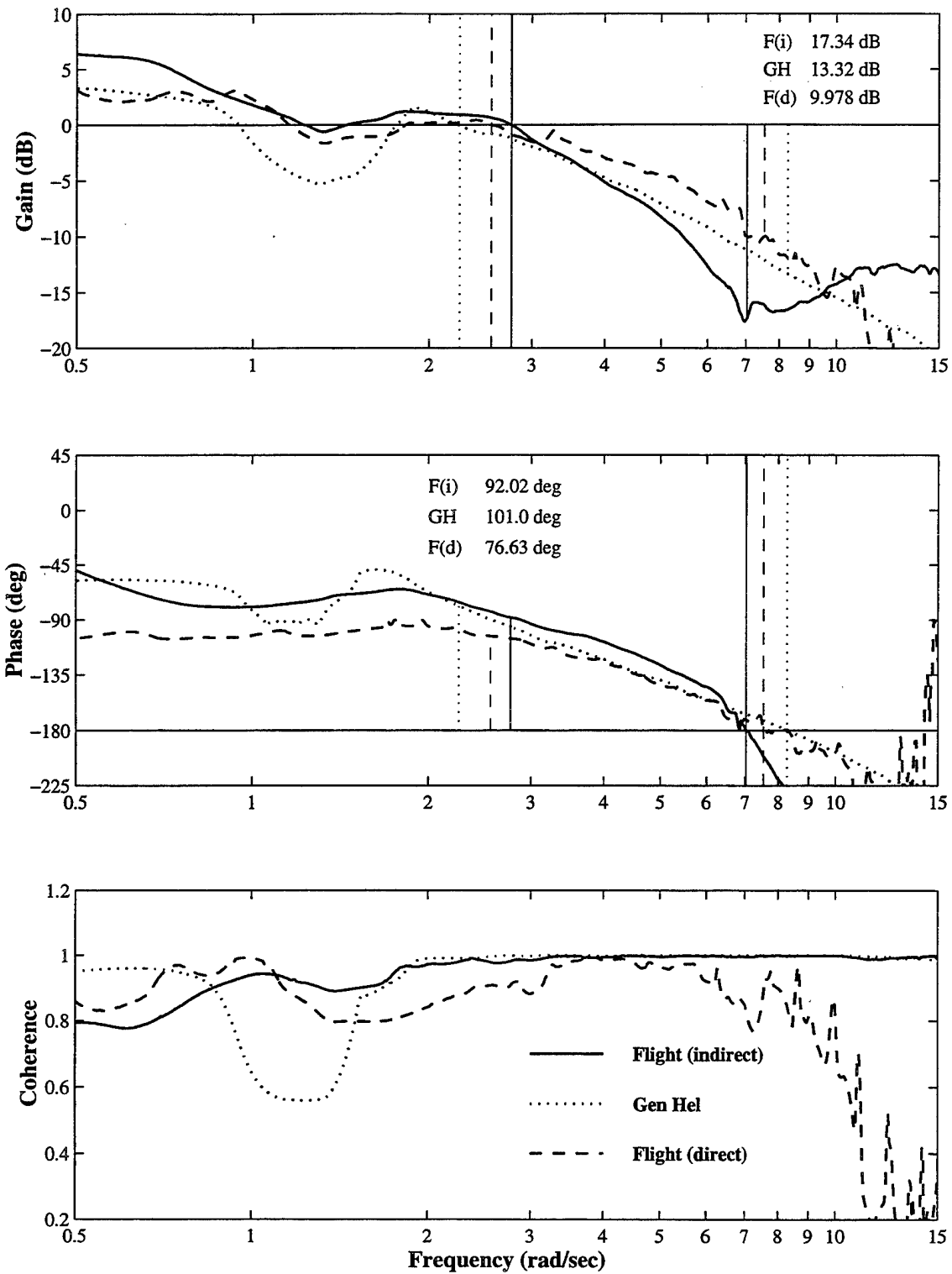


Figure F.8. Continued, (e) 70 Knots

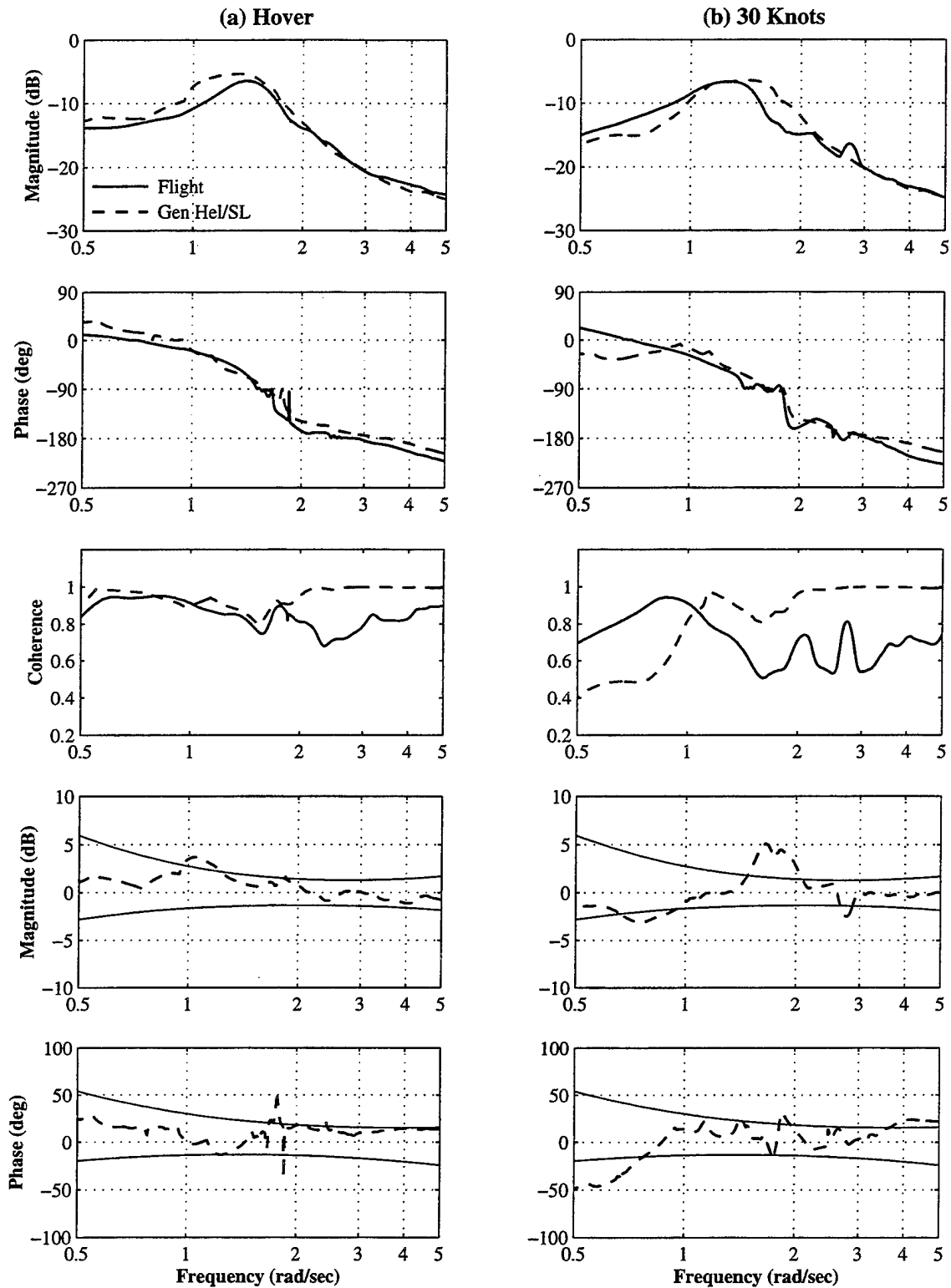


Figure F.9. 4K CONEX Load Motion, Lateral Axis, (a) Hover, (b) 30 Knots

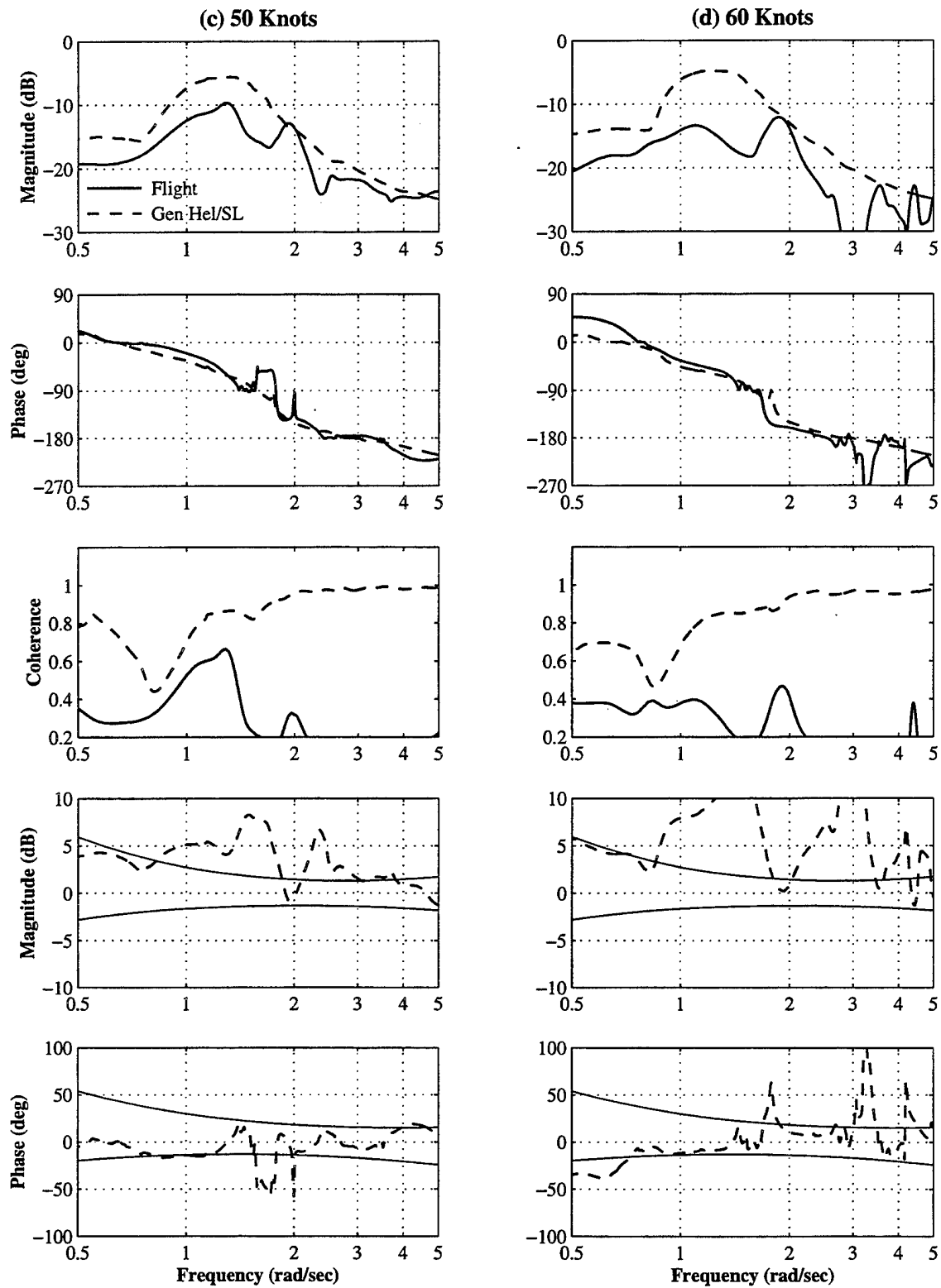


Figure F.9. Continued, (c) 50 Knots, (d) 60 Knots

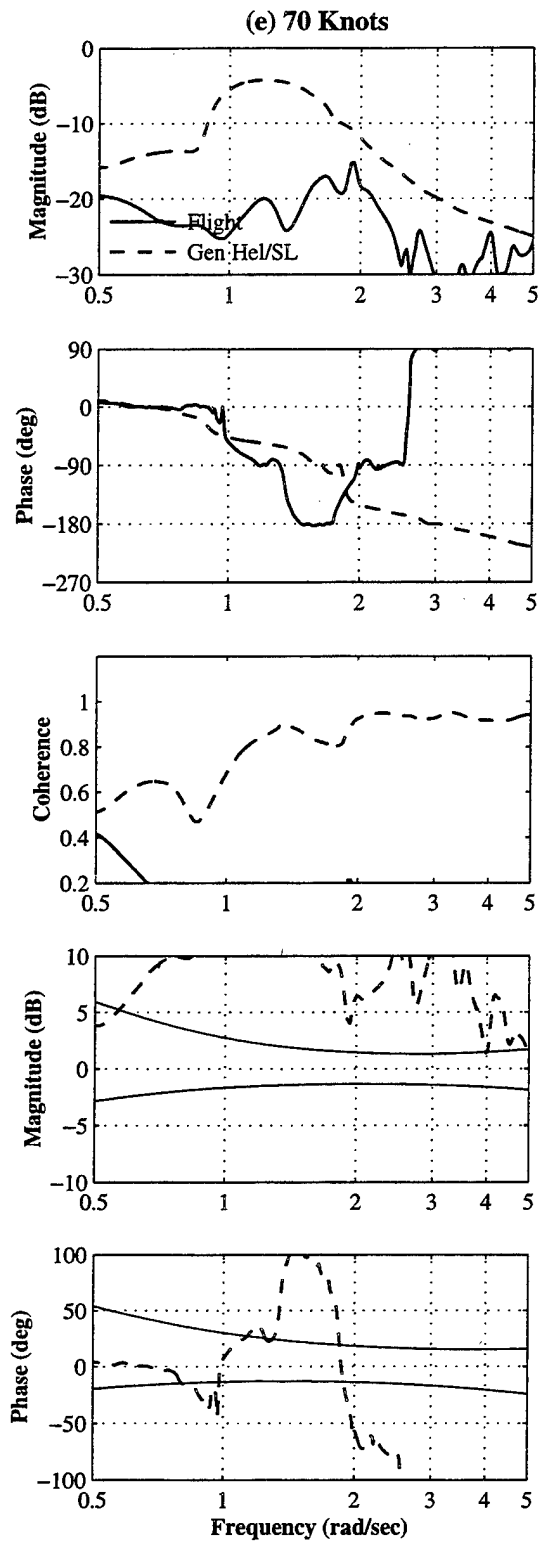


Figure F.9. Continued, (e) 70 Knots

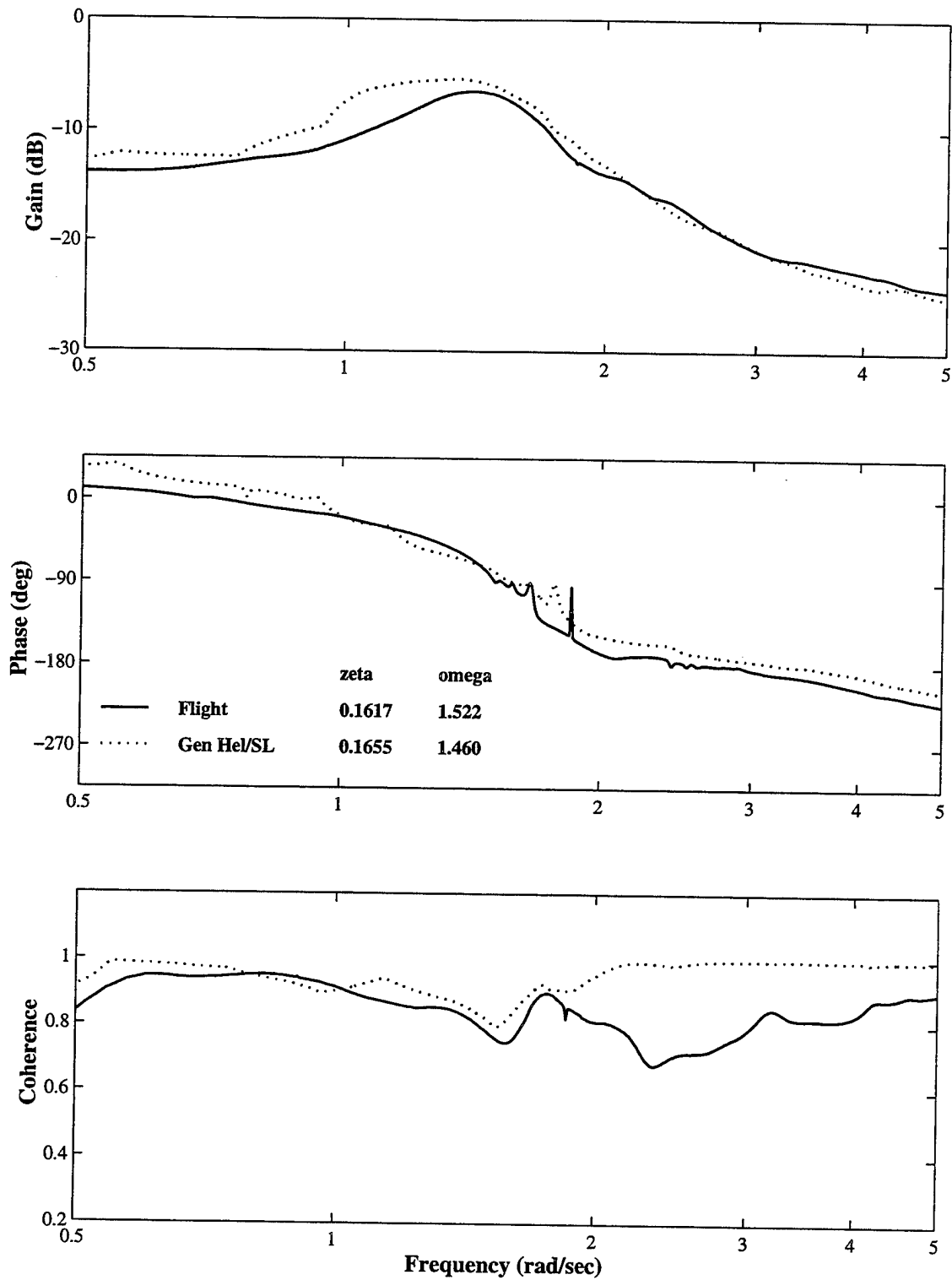


Figure F.10. 4K CONEX Load Motion Determination, Lateral Axis, (a) Hover



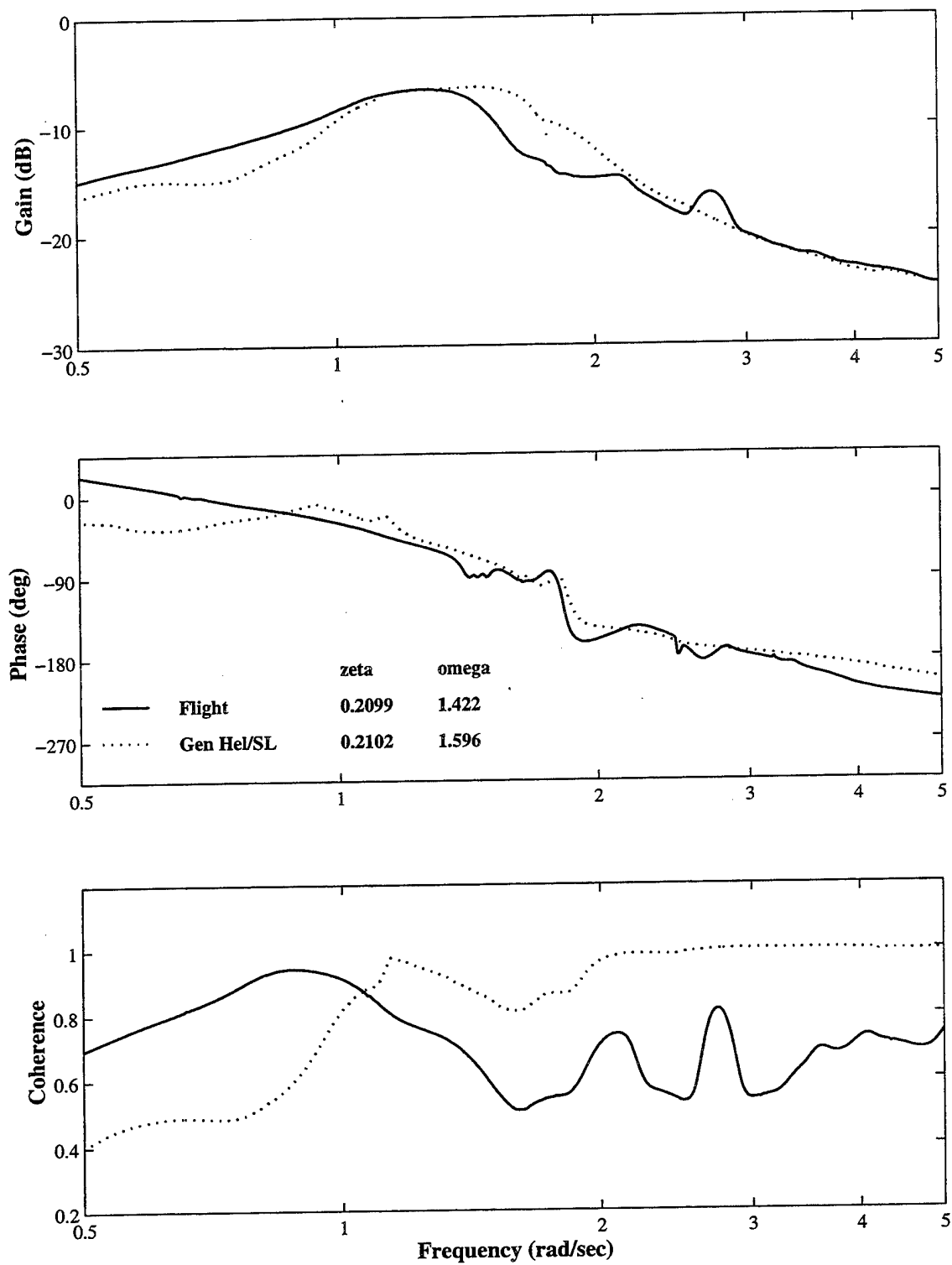


Figure F.10. Continued, (b) 30 Knots

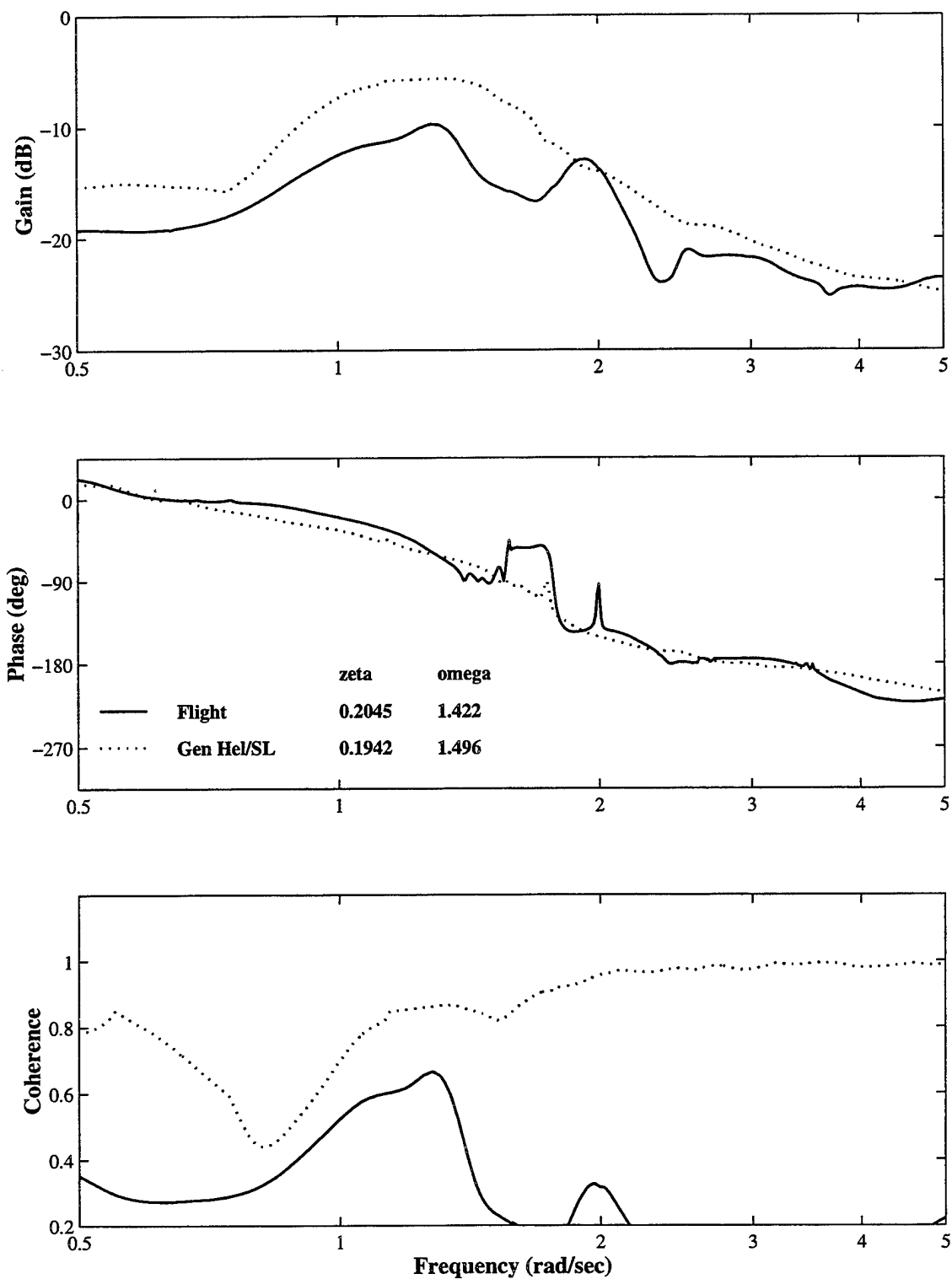


Figure F.10. Continued, (c) 50 Knots

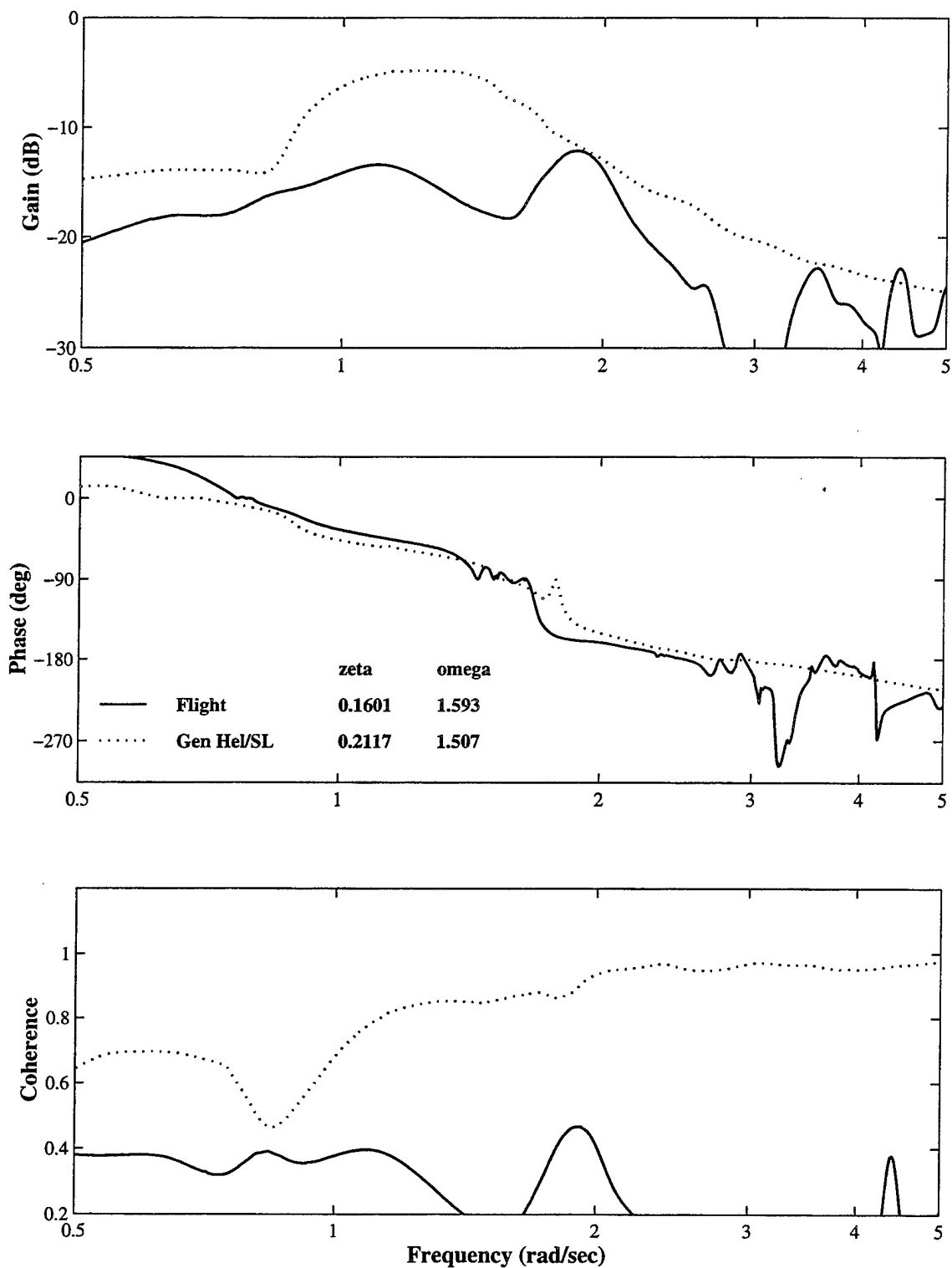


Figure F.10. Continued, (d) 60 Knots

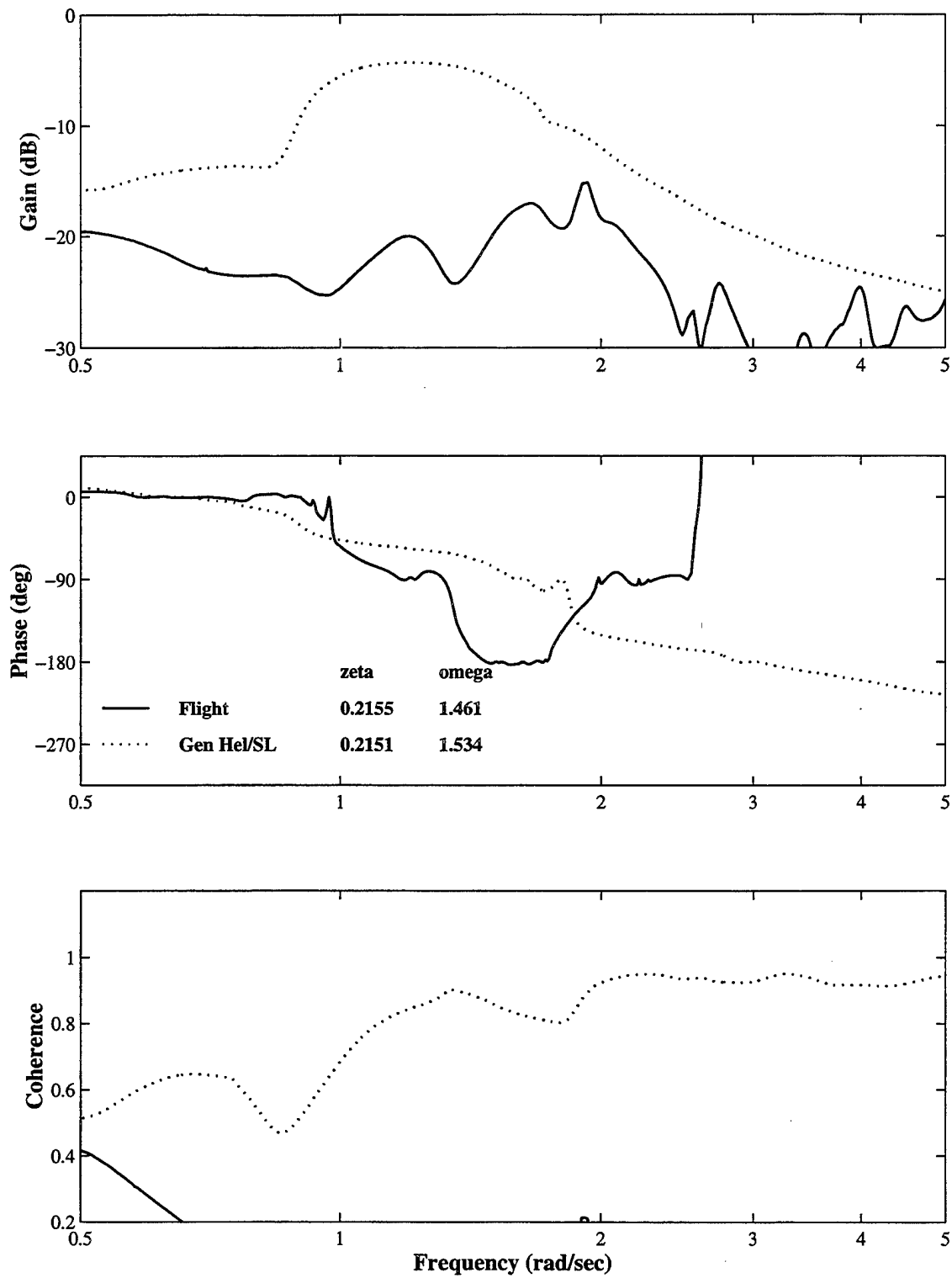


Figure F.10. Continued, (e) 70 Knots

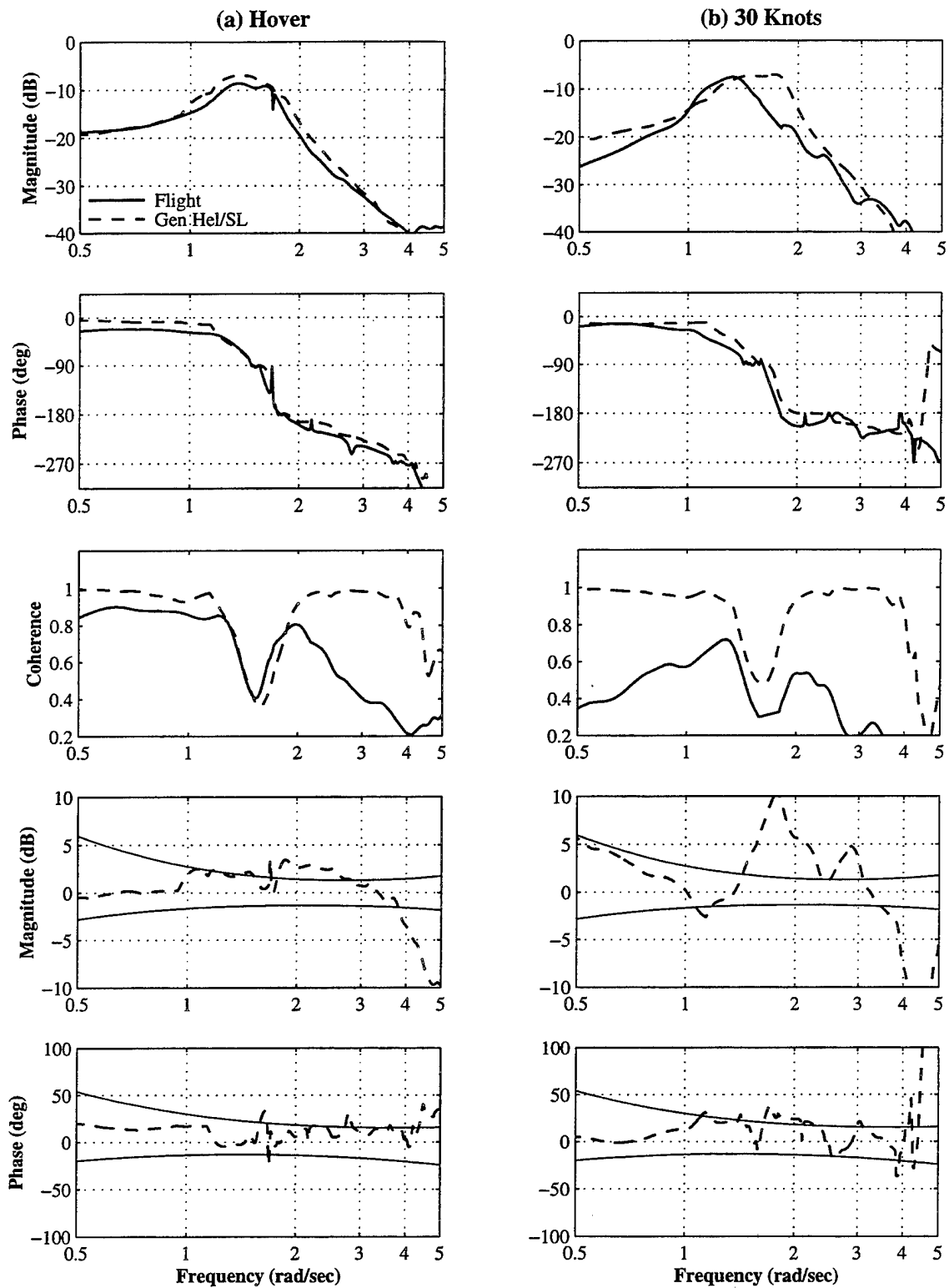


Figure F.11. 4K CONEX Load Motion, Longitudinal Axis, (a) Hover, (b) 30 Knots

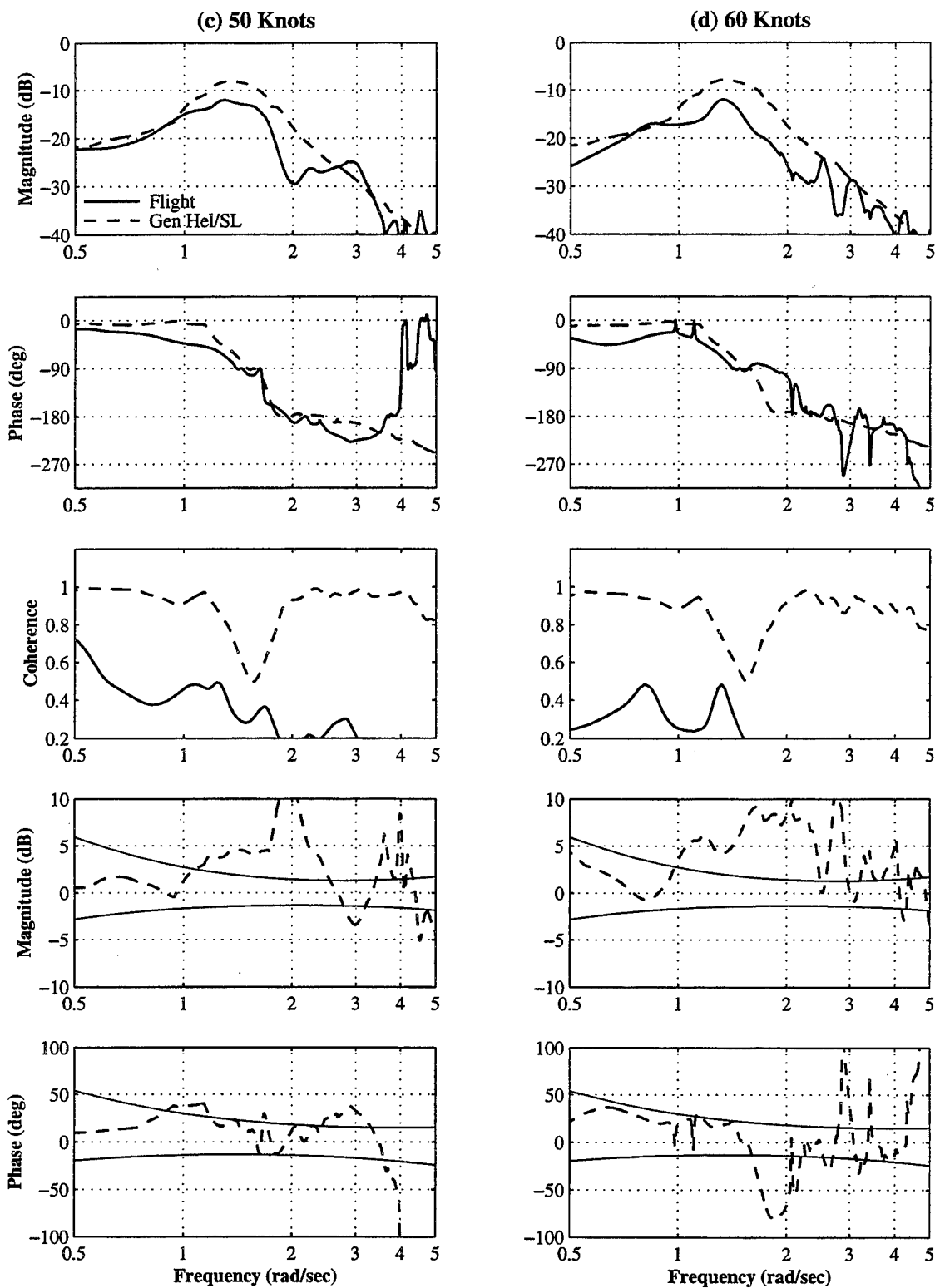


Figure F.11. Continued, (c) 50 Knots, (d) 60 Knots

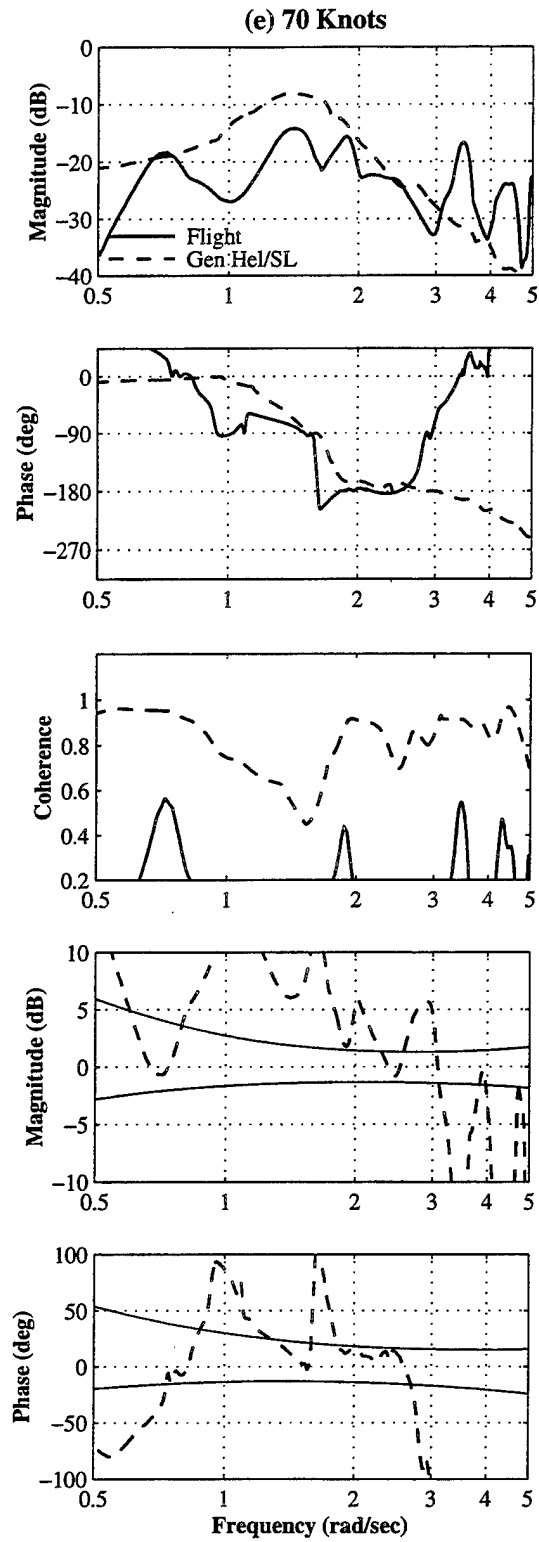


Figure F.11. Continued, (e) 70 Knots

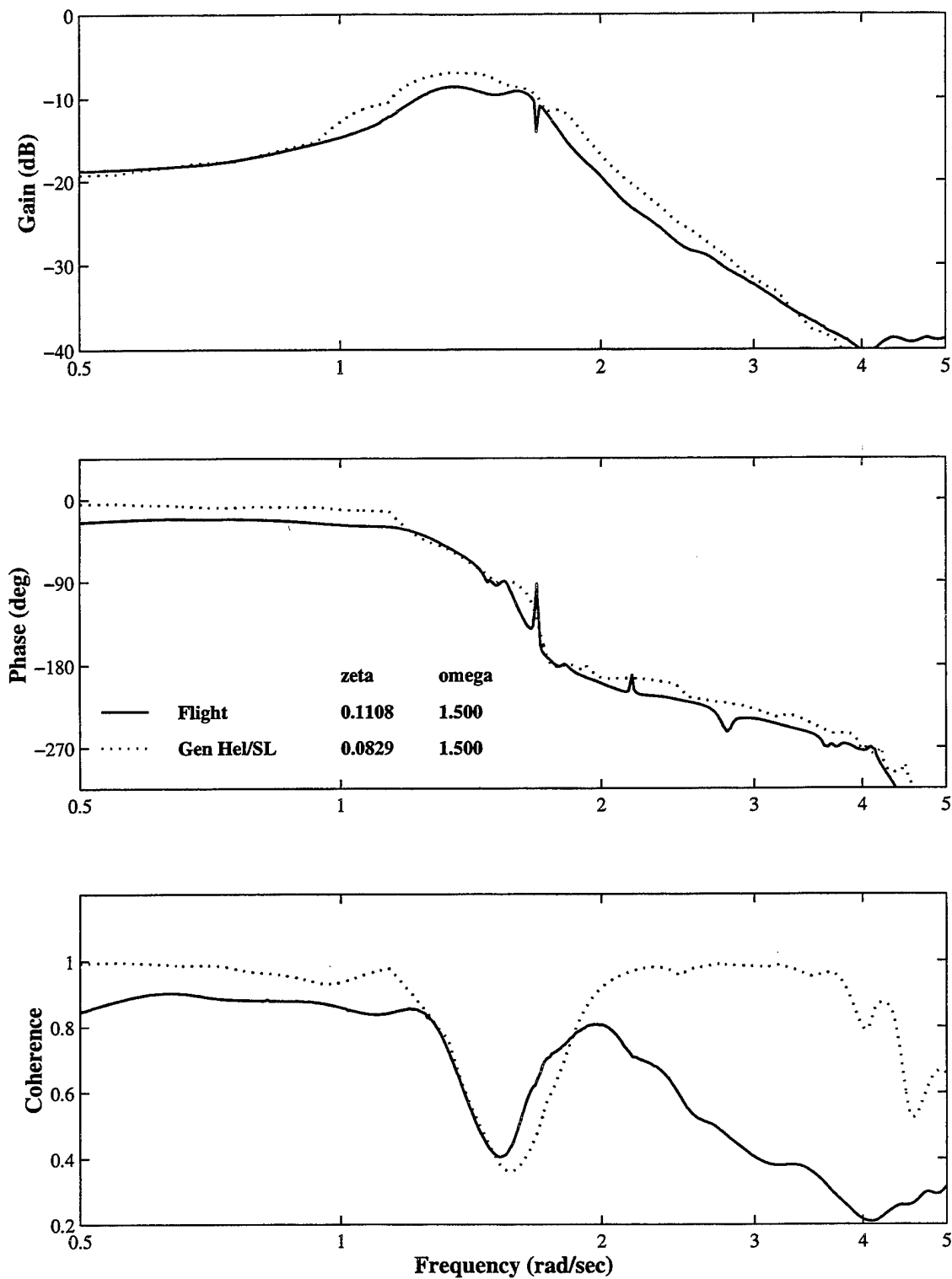


Figure F.12. 4K CONEX Load Motion Determination, Longitudinal Axis, (a) Hover



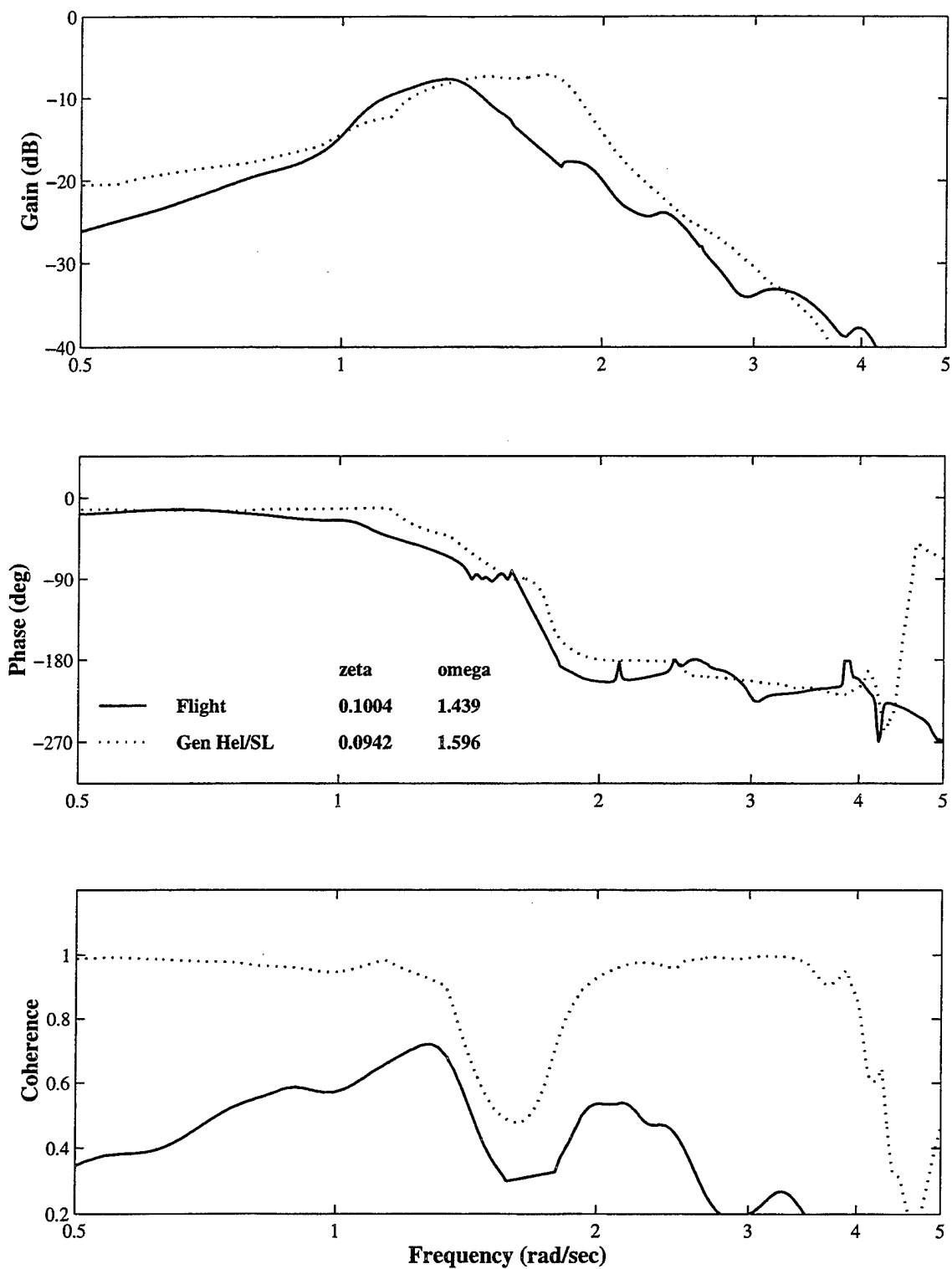


Figure F.12. Continued, (b) 30 Knots

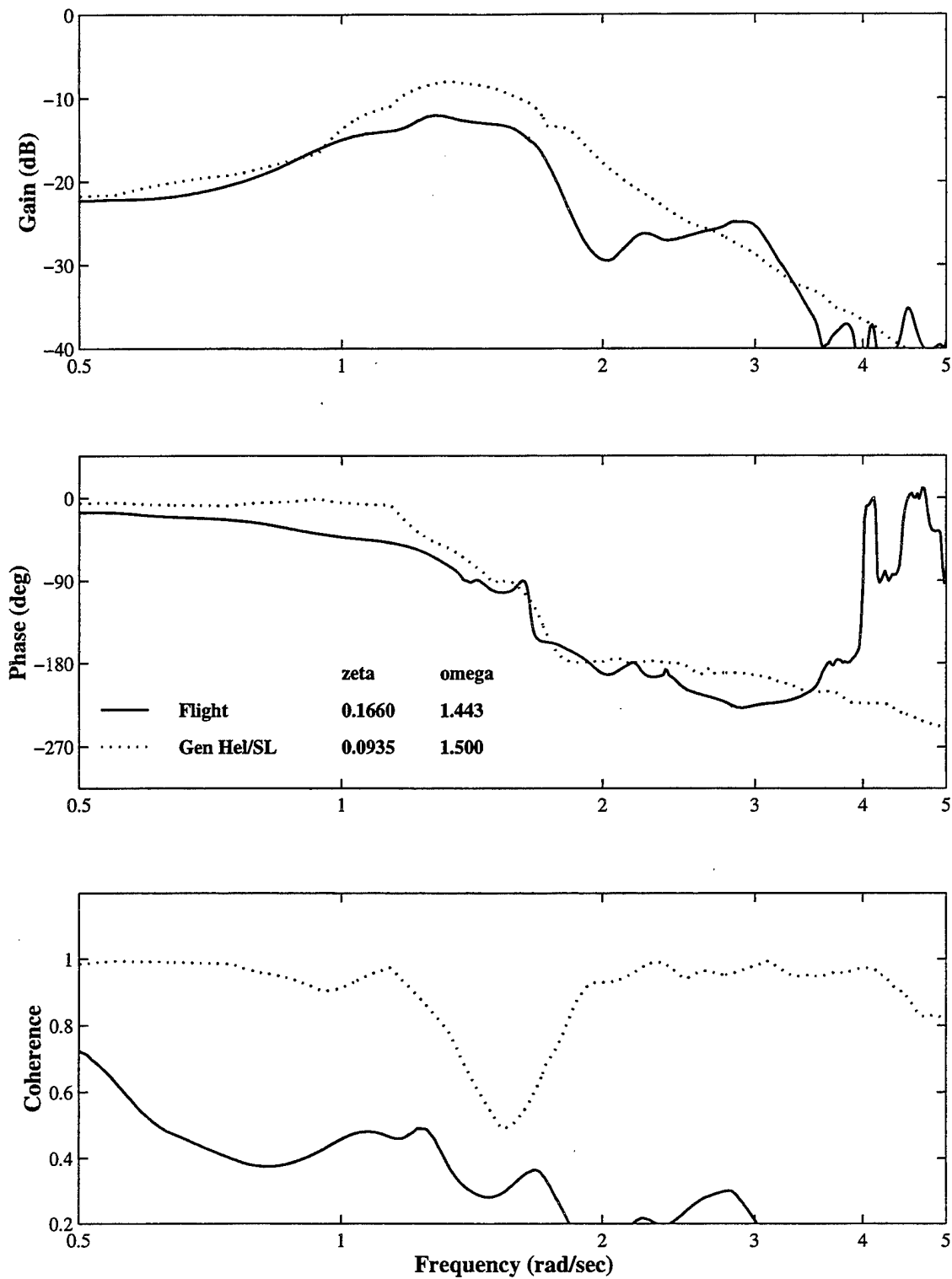


Figure F.12. Continued, (c) 50 Knots

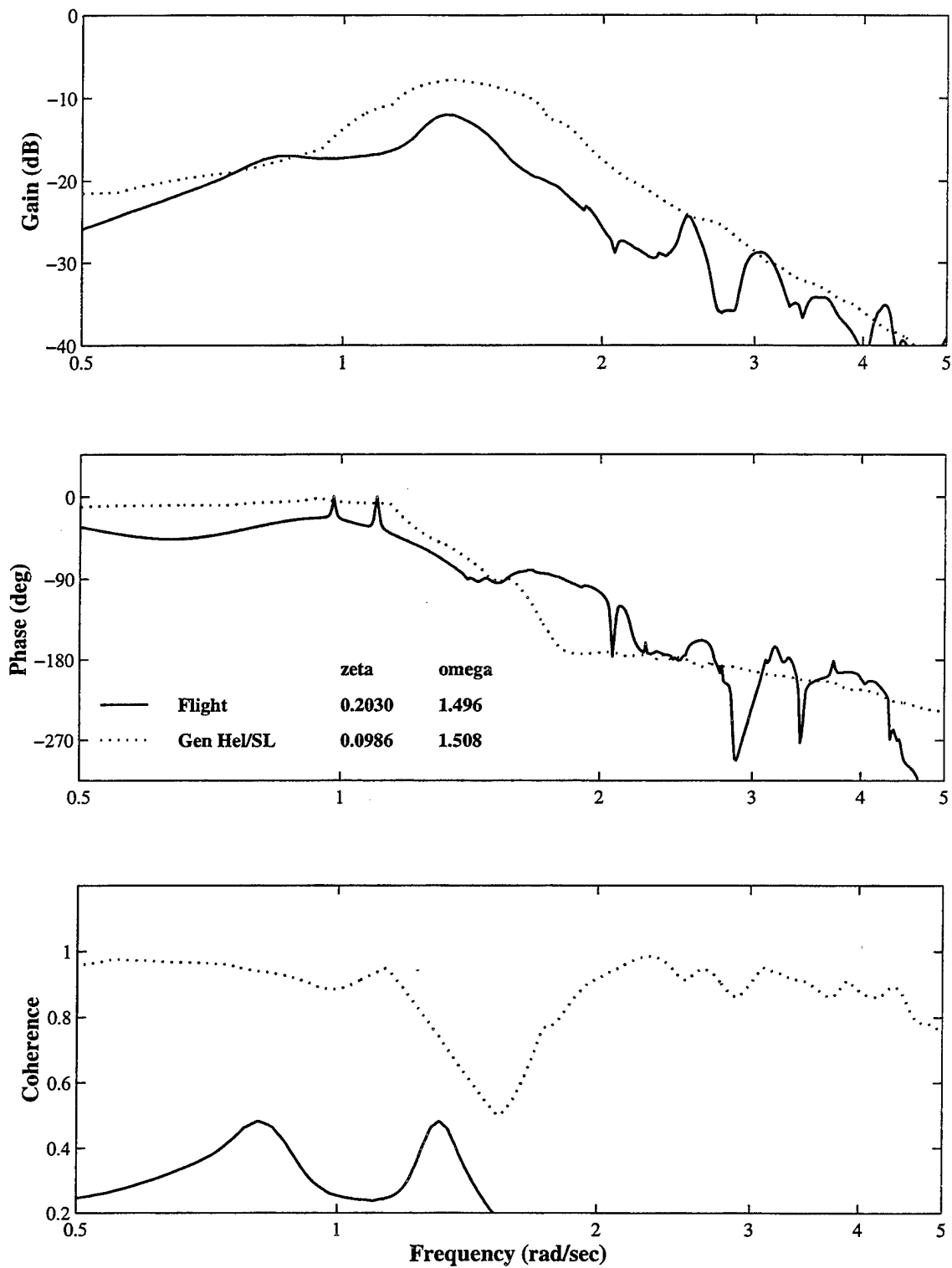


Figure F.12. Continued, (d) 60 Knots

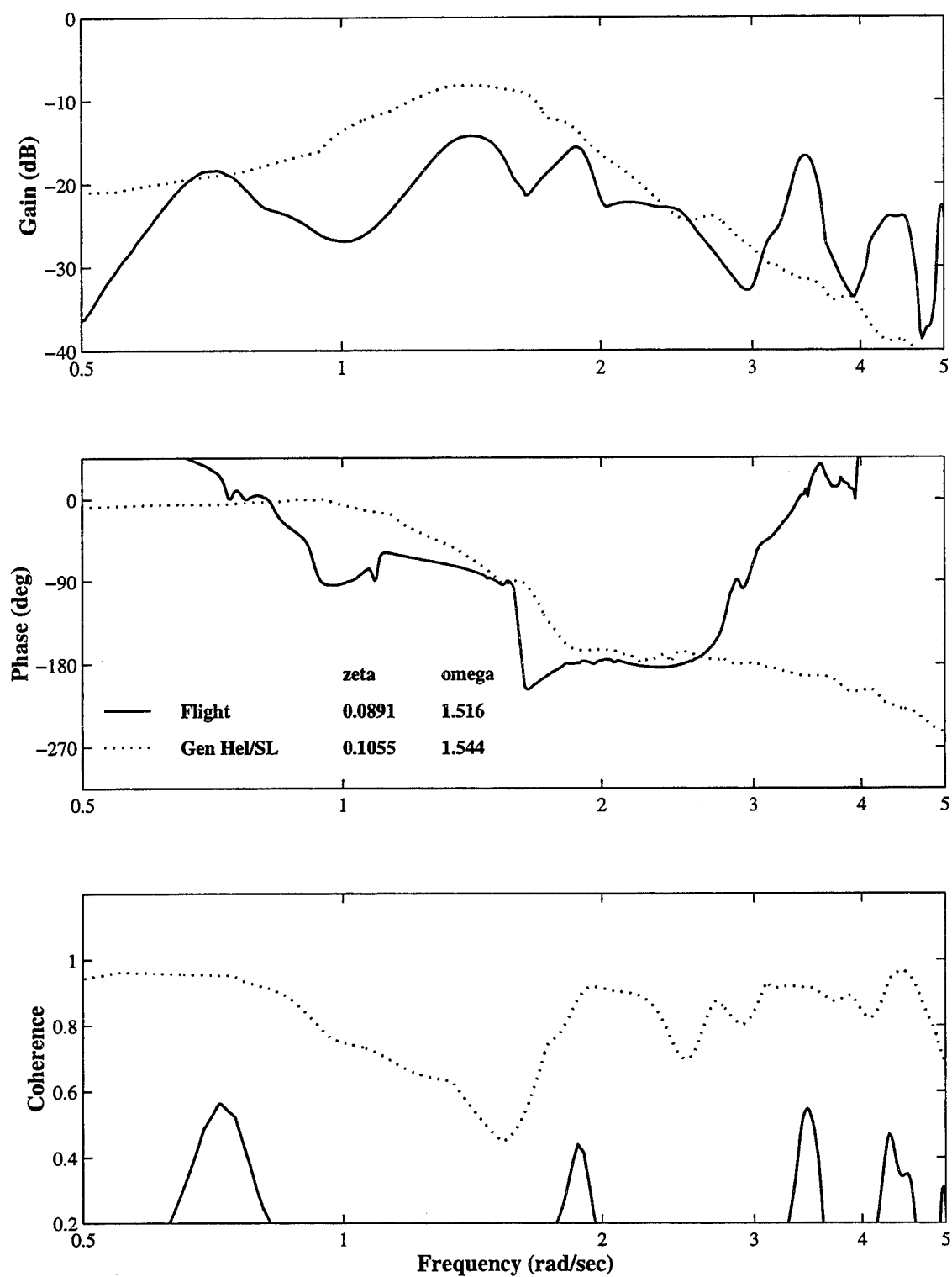


Figure F.12. Continued, (e) 70 Knots



## REFERENCES

1. *Handling Qualities Requirements for Military Rotorcraft*, Aeronautical Design Standard (ADS) 33D, United States Army Aviation and Troop Command (USAATC), St. Louis, MO, July 1994.
2. Robinson, C. S., Wood, E. R., King, R. L., *Simulation of Helicopter Dynamic Mechanical Instability by MAPLE<sup>®</sup> Based Nonlinear LaGrangian Derivation*, AIAA-98-2005, 39<sup>th</sup> Structures, Structural Dynamics, and Materials Conference, Long Beach, CA, April 20-23, 1998.
3. Tischler, Mark B., and Kuritsky, Avraham, eds., *Ten Years of Cooperation, The U.S./Israel Memorandum of Agreement for Cooperative Research on Rotorcraft Aeromechanics and Man-Machine Integration Technology, 1986-1996*, USAATC and M.O.D.-Directorate of Research and Development (MAFAT), Aeroflightdynamics Directorate (AVRDEC), Moffett Field, CA, October 1996.
4. Kuritsky, Avraham, ed., *U.S./Israel MOA on Rotorcraft Aeromechanics and Man-Machine Integration Technology, 23<sup>rd</sup> Meeting, 26-29 October 1998*, MAFAT, Israel, October 1998.
5. McCoy, Allen H., *Flight Testing and Real-Time System Identification Analysis of a UH-60A Black Hawk Helicopter With an Instrumented External Sling Load*, NASA Contractor Report 1998-196710, June 1998.
6. Cicolani, Luigi S., Tyson, Peter H, Blanken, Chris, *Flight Test Plan*, Army/NASA Rotorcraft Division, Aeroflightdynamics Directorate, Ames Research Center, Moffett Field, CA, September 1998.
7. *Operator's Manual for Army Models UH-60A, UH-60L, and EH-60A Helicopters*, US Army Technical Manual, TM-1-1520-237-10, 31 October, 1996.
8. Prouty, R. W., *Helicopter Performance, Stability, and Control*, Robert E. Krieger Publishing Company, 1990.
9. *Multiservice Helicopter External Air Transport: Vols I, II, III*, US Army FM-55-450-3,4,5, February 1991.
10. Tischler, Mark B., *System Identification Methods for Aircraft Flight Control Development and Validation*, NASA Technical Memorandum 110369, USAATCOM Technical Report 95-A-007, October 1995.
11. Williams, Jeffery N., Ham, Johnnie A., Tischler, Mark B., *Flight Test Manual, Rotorcraft Frequency Domain Testing*, AQTG Project No. 93-14, United States Army Aviation

---

Technical Test Center, Airworthiness Qualification Test Directorate, Edwards AFB, CA, September 1995.

12. Cicolani, Luigi S., Kanning, Gerd, *Equations of Motion of Slung-Load Systems, Including Multilift Systems*, NASA Technical Paper 3280, November 1992.
13. Howlett, J. J., *UH-60A Black Hawk Engineering Simulation Program: Volume I - Mathematical Model and Volume II - Background Report*, NASA Contractor Report 166309 and 166310, Sikorsky Aircraft Document No. SER 70452 and 70602, December 1981.
14. Ballin, Mark G., Dalang-Secréan, Marie-Alix, *Validation of the Dynamic Response of a Blade-Element UH-60 Simulation Model in Hovering Flight*, American Helicopter Society 46th Annual Forum, May 1990.
15. Ballin, Mark G., *Validation of a Real-Time Engineering Simulation of the UH-60A Helicopter*, NASA Technical Memorandum 88360, February 1987.
16. Pitt, D. M. and Peters, D. A., *Theoretical Prediction of Dynamic Inflow Derivatives*, "Vertica", Volume 5, 1981, 21-34.
17. Bailey, Jr., F. J., *A Simplified Theoretical Method of Determining the Characteristics of a Lifting Rotor in Forward Flight*, NACA Report 716, 1941.
18. Ronen, Tuvya, *Dynamics of a Helicopter With a Sling Load*, PhD Dissertation, Department of Aeronautics and Astronautics, Stanford University, Palo Alto, CA, August 1985.
19. McKee, John W., Naeseth, Rodger L., *Experimental Investigation of the Drag of Flat Plates and Cylinders in the Slipstream of a Hovering Rotor*, NACA Technical Note 4239, April 1958.
20. Boatwright, Daniel W., *Measurements of Velocity Components in the Wake of a Full-Scale Helicopter Rotor in Hover*, USAAMRDL Technical Report 72-33, Fort Eustis, Virginia, August 1972.
21. Laub, G. H., Kodani, H. M., *Wind Tunnel Investigation of Aerodynamic Characteristics of Scale Models of Three Rectangular Cargo Containers*, NASA Technical Memorandum X-62169, July 1972.
22. Laub, G. H., Kodani, H. M., *Wind Tunnel Investigation of Aerodynamic Characteristics of a Scale Model of a D5 Bulldozer and M109 Self-Propelled 155 mm Howitzer*, NASA Technical Memorandum X-62330, January 1974.
23. Sheldon, D. F., *An Appreciation of the Dynamic Problems Associated with the External Transportation of Loads from a Helicopter - State of the Art*, Vertica, Vol. 1, 1977.

- 
24. Watkins, T. C., Sinacori, J.B., Kessler, D. F., *Stabilization of Externally Slung Helicopter Loads*, USAAMRDL TR-74-42, August 1974.
  25. Simpson, A., Flower, J.W., *Unsteady Aerodynamics of Oscillating Containers and Application to the Problem of Dynamics Stability of Helicopter Underslung Loads*, AGARD-CP-235, May 1978.
  26. Miller, D. G., Lu, Y., White, F., Osciak, E., Roberts, B., Price., Weidorn, J., *Flight Simulation as a Tool to Develop V-22 Slung Load Capabilites*, AHS 55<sup>th</sup> Annual Forum, May 1999.
  27. Mansur, M. Hossein, Tischler, Mark B., *An Empirical Correction Method for Improving off-Axes Response Prediction in Component Type Flight Mechanics Helicopter Models*, NASA Technical Memorandum 110406, USAATCOM Technical Report 96-A-010, February 1997.
  28. Curtiss, H. C., Jr., *On the Calculation of the Response of Helicopters to Control Inputs*, Paper No. F07, 18<sup>th</sup> European Rotorcraft Forum, Avignon, France, September 14-17, 1992.
  29. Lawrence, Thomas H., Gerdes, Walter H., Yakzan, Samir S., *Use of Simulation for Qualification of Helicopter External Loads*, AHS 50<sup>th</sup> Annual Forum, Washington, D.C., May 11-13, 1994.
  30. Etkin, Bernard, *Dynamics of Flight – Stability and Control, Second Edition*, John Wiley & Sons, New York, 1982.





## INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center .....2  
8725 John J. Kingman Rd., STE 0944  
Ft. Belvoir, Virginia 22060-6218
  
2. Dudley Knox Library .....2  
Naval Postgraduate School  
411 Dyer Rd.  
Monterey, California 93943-5101
  
3. Dr. E. Roberts Wood .....2  
Department of Aeronautics and Astronautics  
Code: AA/WD  
Naval Postgraduate School  
411 Dyer Rd.  
Monterey, California 93943-5101
  
4. Dr. Mark B. Tischler .....2  
U.S. Army/NASA Rotorcraft Division  
Ames Research Center  
Mail Stop: 210-5  
Moffett Field, California 94035-1000
  
5. Dr. Garth V. Hobson .....1  
Department of Aeronautics and Astronautics  
Code: AA/HG  
Naval Postgraduate School  
411 Dyer Rd.  
Monterey, California 93943-5101
  
6. LT Peter H. Tyson .....1  
908 Fernhill Rd.  
Glenside, Pennsylvania 19038
  
7. Mr. Luigi Cicolani .....1  
U.S. Army/NASA Rotorcraft Division  
Ames Research Center  
Mail Stop: 210-5  
Moffett Field, California 94035-1000
  
8. Mr. Barry Lakin Smith .....1  
U.S. Army/NASA Rotorcraft Division  
Ames Research Center  
Mail Stop: 210-5  
Moffett Field, California 94035-1000

9. Mr. Andy Kerr..... 1  
U.S. Army/NASA Rotorcraft Division  
Ames Research Center  
Mail Stop: 219-3  
Moffett Field, California 94035-1000
10. Mr. Ed Aiken..... 1  
U.S. Army/NASA Rotorcraft Division  
Ames Research Center  
Mail Stop: 210-1  
Moffett Field, California 94035-1000
11. Mr. Aviv Rosen..... 1  
Technion Israel Institute of Technology  
Department of Aeronautics  
Technion City, Haifa, 32000, Israel